DSグラフィックス入門

Ver 2.4.0

任天堂株式会社発行

目次

表

図

改訂履歴

版	改訂日	改 訂 内 容
2.4.0	2009-03-10	・TWL 対応に伴い、タイトルを「DS グラフィックス入門」に変更 ・ハードウェア構成などの内容を TWL のものに変更 ・見出しおよび文中の NITRO 表記を削除または DS に変更 ・「7.3 グラフィックス表示モード使用例 3 ＜2 画面での 3D 表示＞」を最新のサンプルデモの内容に合わせて更新
2.3.0	2004-09-13	・「4.2.7 補足 1:ポリゴンのカラーの決め方」の解説に注意事項を加筆
2.2.0	2004-06-21	・「3.4 カラーパレット」 拡張パレットの解説を修正 ・「4.2.4 ライト計算」のスペキュラ成分計算式を修正 ・「4.3.9 アンチエイリアス」の解説に補足説明を追加
2.1.0	2004-03-23	「2.3 NITRO 表示システムブロック図」を追加 ・「7 付録」を追加
2.0.0	2004-01-28	・リリース

1 はじめに

本解説書は、DS(TWL および NITRO)のグラフィックスの機能を“紹介する”ことを目的として書かれています。

よって、各機能を実際のゲーム中で使用した場合の以下の点については、詳しく言及はしていません。

- ROM や RAM をどれくらい消費するか？
- 各プロセッサの処理時間がどれくらいかかるか？
- 各機能の「使いやすさ」や「実現しやすさ」について。
- 実際のプログラムコードについて。

なお、本解説書の内容は、『TWL プログラミングマニュアル V1.4』に基づいて記述しています。

2 DSグラフィックスハードウェア概要

2.1 ハードウェア構成

DS グラフィックスにおける主要なハードウェア構成を示します。

■ CPU コア

プロセッサに ARM9 と ARM7 と呼ばれる 2 つの CPU コアを搭載しています。

- ARM9 ・・・クロック周波数 67.028 MHz（TWL の倍速モード時は 134.056 MHz）
- ARM7 ・・・クロック周波数 33.514 MHz

■ グラフィックスエンジン

2D グラフィックスと 3D グラフィックスのそれぞれを制御するグラフィックスエンジンがあります。
TWL のグラフィックスエンジンは、NITRO での不具合を修正した回路の使用／不使用を制御できることを除いては、特に仕様の変更はありません。

- ・2D グラフィックスエンジン A
- ・2D グラフィックスエンジン B

… 2D グラフィックスを制御します。（クロック周波数 33.514 MHz）

- ・3D グラフィックスエンジン
  - ・GE（ジオメトリエンジン） … 3D グラフィックスの計算処理を制御します。（クロック周波数 33.514 MHz）
  - ・RE（レンダリングエンジン） … 3D グラフィックスの描画処理を制御します。（クロック周波数 33.514 MHz）

＊2D グラフィックスエンジン B は、適用できる 2D グラフィックス機能に制限があります。

■　メモリ

使用できるメモリには、大きく分けて以下の 3 種類があります。

それぞれのメモリの詳細と容量は次の 表 2-1のようになっています。

- メインメモリ
- VRAM（VRAM-A～VRAM-I の 9 つがあります）
- ワーク RAM（NITRO 互換モード時は従来通りに ARM9，ARM7 共用と ARM7 専用の 2 種類、TWL モード時はそれに ARM9，ARM7，DSP 命令 RAM 共用、ARM9，ARM7，DSP データ RAM 共用を加えた 4 種類があります）

表 2-1　メモリと容量

<table>
<tr><th colspan="2">メモリ</th><th colspan="2">容量</th></tr>
<tr><td colspan="2">メインメモリ</td><td colspan="2">16M バイト（NITRO 互換モード時は 4M バイト）</td></tr>
<tr><td rowspan="9">VRAM</td><td>VRAM-A</td><td>128K バイト</td><td rowspan="4">合計
512K バイト</td></tr>
<tr><td>VRAM-B</td><td>128K バイト</td></tr>
<tr><td>VRAM-C</td><td>128K バイト</td></tr>
<tr><td>VRAM-D</td><td>128K バイト</td></tr>
<tr><td>VRAM-E</td><td colspan="2">64K バイト</td></tr>
<tr><td>VRAM-F</td><td>16K バイト</td><td rowspan="2">合計
32K バイト</td></tr>
<tr><td>VRAM-G</td><td>16K バイト</td></tr>
<tr><td>VRAM-H</td><td>32K バイト</td><td rowspan="2">合計
48K バイト</td></tr>
<tr><td>VRAM-I</td><td>16K バイト</td></tr>
<tr><td rowspan="2">ワーク RAM
（NITRO 互換）</td><td>ARM9，ARM7 共用</td><td colspan="2">32K バイト</td></tr>
<tr><td>ARM7 専用</td><td colspan="2">64K バイト</td></tr>
<tr><td rowspan="4">ワーク RAM
（TWL）</td><td>ARM9，ARM7 共用</td><td colspan="2">32K バイト+256K バイト （合計 288K バイト）</td></tr>
<tr><td>ARM7 専用</td><td colspan="2">64K バイト</td></tr>
<tr><td>ARM9，ARM7，DSP 命令 RAM 共用</td><td colspan="2">256K バイト</td></tr>
<tr><td>ARM9，ARM7，DSP データ RAM 共用</td><td colspan="2">256K バイト</td></tr>
</table>

【参考】

AGB で「CPU 外部ワーク RAM」と呼ばれていたメモリは『メインメモリ』に、
AGB で「CPU 内部ワーク RAM」と呼ばれていたメモリは『ワーク RAM』に相当します。

■　LCD

メインLCDとサブLCDの 2 つのLCDがあります。

LCD表示サイズはメインLCD、サブLCD共に 256×192 となっています。（図 2-1）

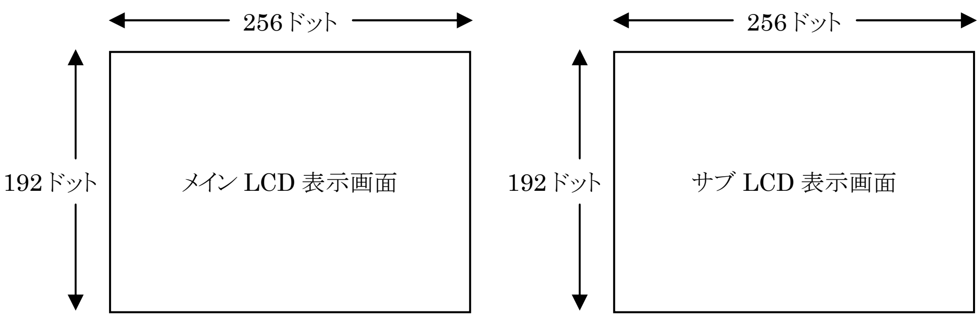


図 2-1　LCD の表示サイズ

2.2 表示モード

LCD に画像を表示するための 3 つの表示モードがあります。

- グラフィックス表示モード
- VRAM 表示モード
- メインメモリ表示モード

○ グラフィックス表示モード

2D グラフィックス機能や 3D グラフィックス機能によって生成された画像を表示するモードです。

○ VRAM 表示モード

VRAM 上に用意された画像データを表示するモードです。
表示に使用する VRAM を VRAM-A ～ VRAM-D から選択することができます。

○ メインメモリ表示モード

メインメモリ上に用意された画像データを表示するモードです。

2.3 表示システムブロック図

図 2-2は表示システムブロック図です。図中の番号は以下の説明番号に対応しています。

① 2DグラフィックスエンジンA
OBJおよびBG画像を生成します。
→「3 2Dグラフィックス」で説明しています。

② 2DグラフィックスエンジンB
2DグラフィックスエンジンAと同じく、OBJおよびBG画像を生成します。
→「3 2Dグラフィックス」で説明しています。

③ 3Dグラフィックスエンジン
3D画像を生成します。
→「4 3Dグラフィックス」で説明しています。

④ OBJ
→「3.2 OBJの機能」で説明しています。

⑤ BG
→「3.3 BGの機能」で説明しています。

⑥ 重ね合わせ＆カラー特殊効果
→「3.6 カラー特殊効果」
「5 2D面と 3D面の融合」で説明しています。

⑦ VRAM
キャラクタデータ、スクリーンデータ、パレットデータ、テクスチャデータ、ビットマップデータなどの画像データを保持するためのメモリです。描画時には必要に応じて 2D グラフィックスエンジンおよび 3D グラフィックスエンジンから参照されます。VRAM 表示モードでは VRAM 上のビットマップデータを表示することができます。

⑧ メインメモリ
メインメモリ表示モードでは、メインメモリ上のビットマップデータを表示することができます。

⑨ キャプチャ機能
2DグラフィックスエンジンB以外で生成された画像をVRAM-A～VRAM-Dのいずれかに取り込む機能です。
→「6 表示キャプチャ」で説明しています。

表示システムブロック図

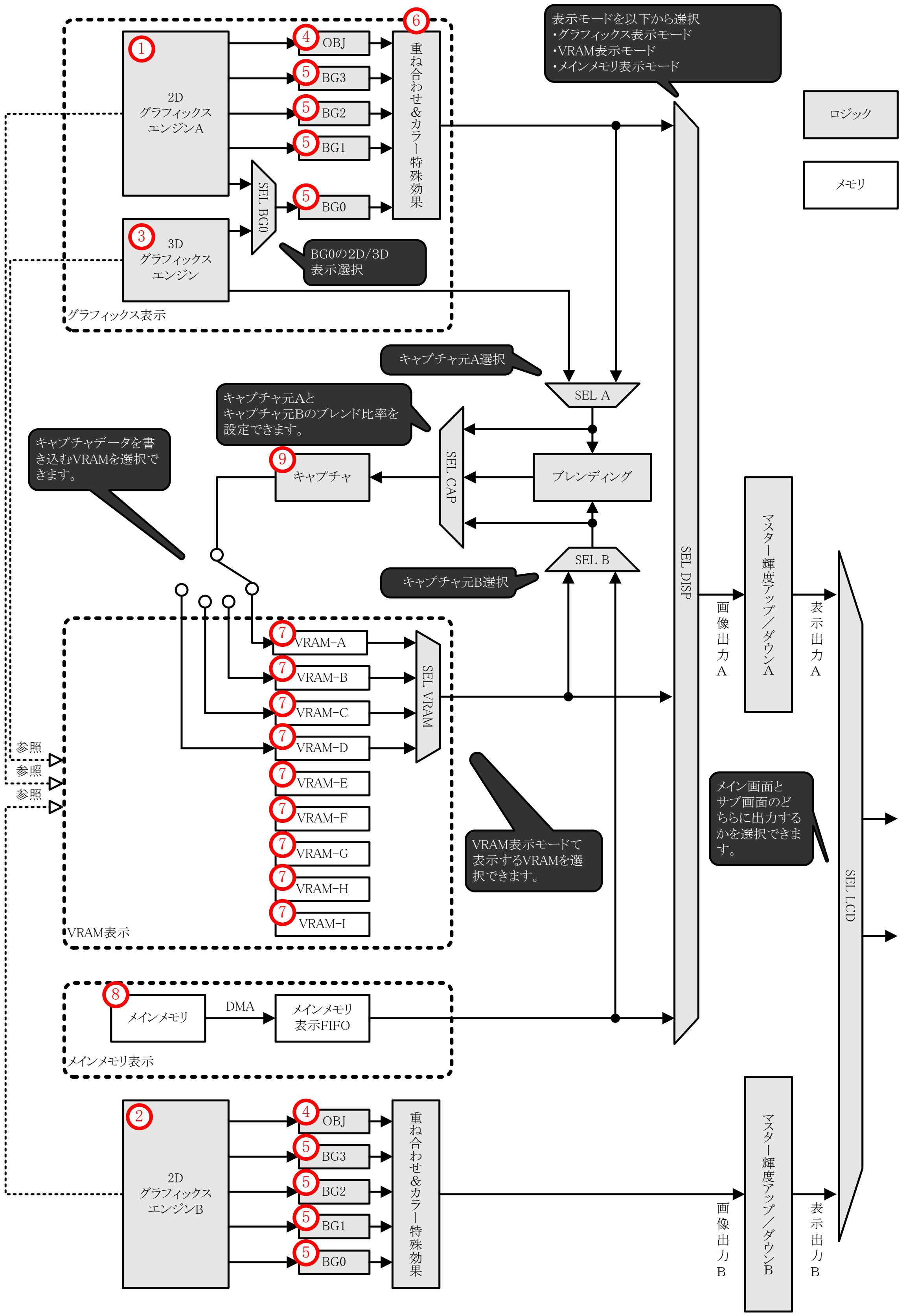


図 2-2 表示システムブロック図

3　2Dグラフィックス

DS の 2D グラフィックス機能は、多くの部分が AGB の表示機能を踏襲しています。

3.1　2Dゲーム画面はどのようにして出来上がっているか？

2Dゲーム画面は大きく分けて、以下の 3 つで構成されています。(図 3-1)

- OBJ（オブジェ）
- BG（ビージー）
- バックドロップ

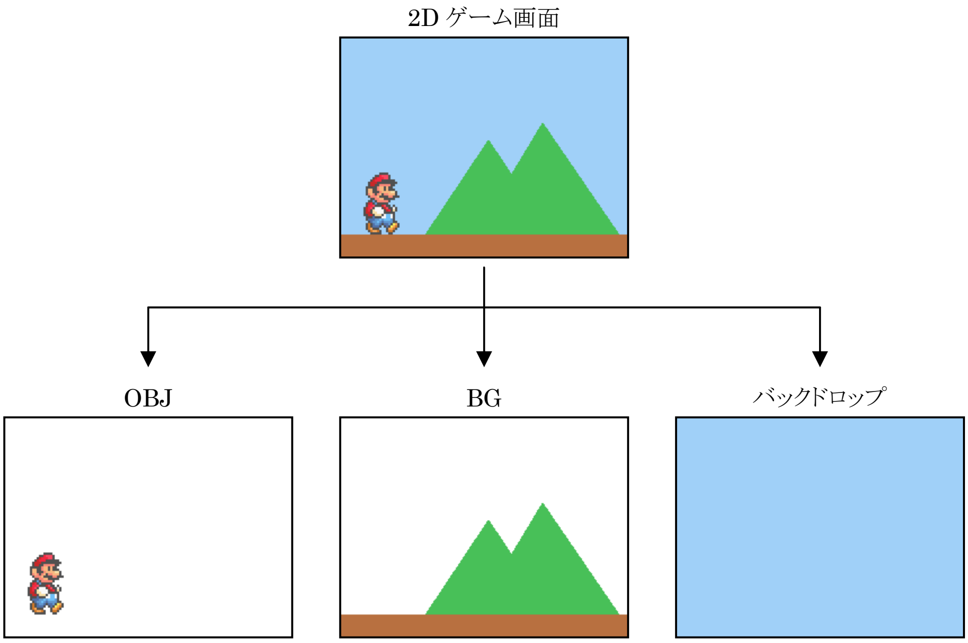


図 3-1 2D ゲーム画面の構成

3.1.1 OBJ(オブジェ)

OBJ は、主に画面内を動き回るもの(プレイヤー、敵、アイテム・・・等)を表示するのに使用します。

OBJは以下に示すような流れで表示されます。(図 3-2)

① 各 OBJ の基となる“キャラクタデータ”を用意します。

② 表示する OBJ 毎に、キャラクタデータ内のどのキャラクタを使用するか、どの位置に表示するか・・・、といったデータを用意します。このデータのことを“OAM(オブジェクトアトリビュートメモリ)データ“といいます。

③ OBJ が描画・表示されます。

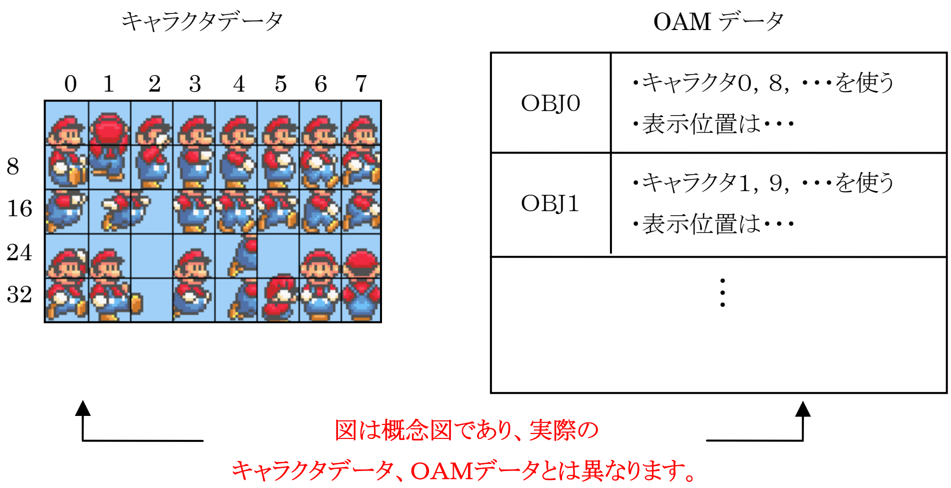


図 3-2 OBJ の表示に必要なデータの概念図

1 描画フレーム内で最大 128 個の OBJ を表示することができます。

OBJ機能の詳細については、「3.2 OBJの機能」で解説します。

3.1.2 BG（ビージー）

BG は、いわゆるゲームの“背景”を表示するのに使用します。

BG を表示するには、以下の 2 種類の方法があります。

1） キャラクタ方式のBG（図 3-3）

① 基となるキャラクタデータを用意します。

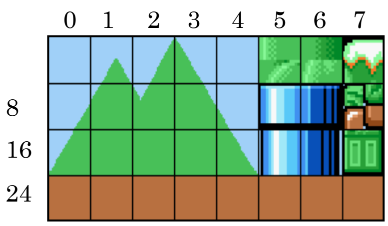
⇩

② どの位置に、何番のキャラクタを配置するかのデータを用意します。このデータを“スクリーンデータ”といいます。

⇩

③ BG が描画・表示されます。

キャラクタデータ



スクリーンデータ

0	0	0	0	0	0	0	0
0	0	0	0	1	2	3	0
0	13	14	8	9	10	11	12
0	21	22	16	17	18	19	20
24	24	24	24	24	24	24	24

図は概念図であり、実際のキャラクタデータ、
スクリーンデータとは異なります。

結果の BG 画像

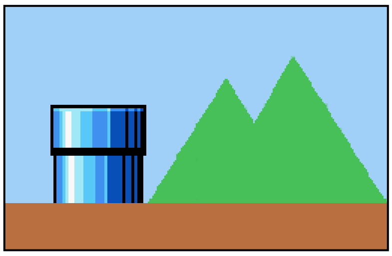


図 3-3 キャラクタ方式の BG の表示に必要なデータの概念図

2) ビットマップ方式の BG

キャラクタデータとスクリーンデータを使用せずに、各ドットの色を直接指定して、BG を描画・表示する方法です。

色の指定の仕方には、カラーパレットを用いる方法と、RGB値を直接指定する方法の 2 種類があります。(図 3-4)

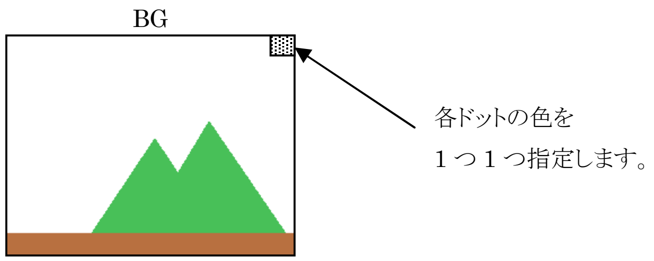


図 3-4 ビットマップ方式の BG

BG機能の詳細については、「3.3 BGの機能」で解説します。

3.1.3 バックドロップ

バックドロップは、画面内の OBJ と BG のどれにも遮られない領域に表示される面です。

バックドロップにはカラーデータのみを指定します。

3.2 OBJの機能

OBJ には、キャラクタ OBJ とビットマップ OBJ の 2 種類があります。

さらにキャラクタ OBJ には、ノーマル OBJ、半透明 OBJ、OBJ ウィンドウの 3 種類があります。

- キャラクタ OBJ
  - ノーマル OBJ
  - 半透明 OBJ
  - OBJ ウィンドウ

  ･･･カラーパレットを使用する OBJ です。
- ビットマップ OBJ ･･･ダイレクトカラーの OBJ です。

【注意】
キャラクタ OBJ、ビットマップ OBJ は、共にキャラクタデータを使用するキャラクタ方式です。
キャラクタ OBJ はキャラクタ方式、ビットマップ OBJ はビットマップ方式というわけではありません。

OBJ毎に設定する項目や適用できる機能は、以下の 表 3-1のようになっています。

表 3-1 OBJ 機能一覧

仕様・機能	キャラクタ OBJ			ビットマップ OBJ
	ノーマル OBJ	半透明 OBJ	OBJ ウィンドウ	
色数／パレット	1） 標準パレット使用時 16 色×16 パレット、 256 色×1 パレット 2） 拡張パレット使用時 16 色×16 パレット(標準パレット) 256 色×16 パレット(拡張パレット)			32768 色
キャラクタ数 (8×8 ドットで換算)	1024～8192 個(16 色モード) 512～4096 個(256 色モード)			256 個、あるいは 1024～2048 個
OBJ サイズ	8×8 16×8 32×8 32×64 8×16 16×16 32×16 64×32 8×32 16×32 32×32 64×64			
1 画面最大表示数	128 個(64×64 ドットで換算)			
1 ライン最大表示数	128 個(8×8 ドットで換算)			
半透明処理	△	○	△	△
ウィンドウ機能	×	×	○	×
回転拡大縮小	○	○	○	○
倍角表示	○	○	○	○
HV フリップ	○	○	○	○
モザイク	○	○	○	○
優先順位	○	○	×	○

○ 色数／パレット
選択できる色数とパレット数です。
ビットマップOBJはカラーパレットを使用しないので、表示色数を表しています。
カラーパレットについては「3.4 カラーパレット」で解説します。

○ キャラクタ数
8×8 ドットのキャラクタを基本キャラクタとした場合に、キャラクタデータ内に定義可能なキャラクタ数です。
VRAM 中にキャラクタデータをどのように配置するかによって、定義可能なキャラクタ数は変わります。

○ OBJ サイズ
OBJ のサイズは、8×8～64×64 の 12 種類から選択することができます。

○ 1 画面最大表示数
1 画面内に表示することができる OBJ の最大数です。

○ 1ライン最大表示数
1 ライン内に表示することができる OBJ の最大数です。

○ 半透明処理
カラー特殊効果処理でのαブレンディングの適用の有無を表します。
表中の△印は、αブレンディングの適用に際して条件を伴うことを意味します。
カラー特殊効果処理については「3.6 カラー特殊効果」で解説します。

○ ウィンドウ機能
ウィンドウ機能については「3.5 ウィンドウ」で解説します。

○ 回転拡大縮小
OBJを回転、拡大、縮小して表示する機能です。(図 3-5)

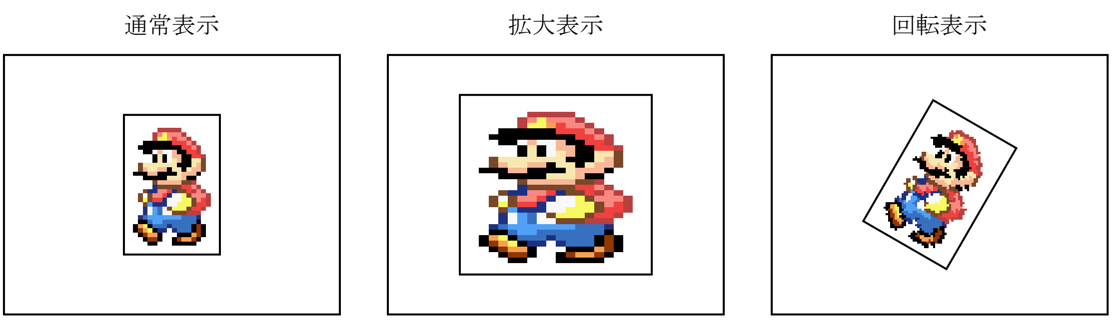


図 3-5 OBJ の回転拡大縮小

＊回転拡大縮小機能を使用している OBJ を、アフィン OBJ と呼びます。

○ 倍角表示

回転拡大縮小機能を使用すると、OBJサイズを超えた領域は欠けて表示されます。

倍角表示機能を使用することで、OBJサイズの 2 倍の領域までなら欠けずに表示することができます。(図 3-6)

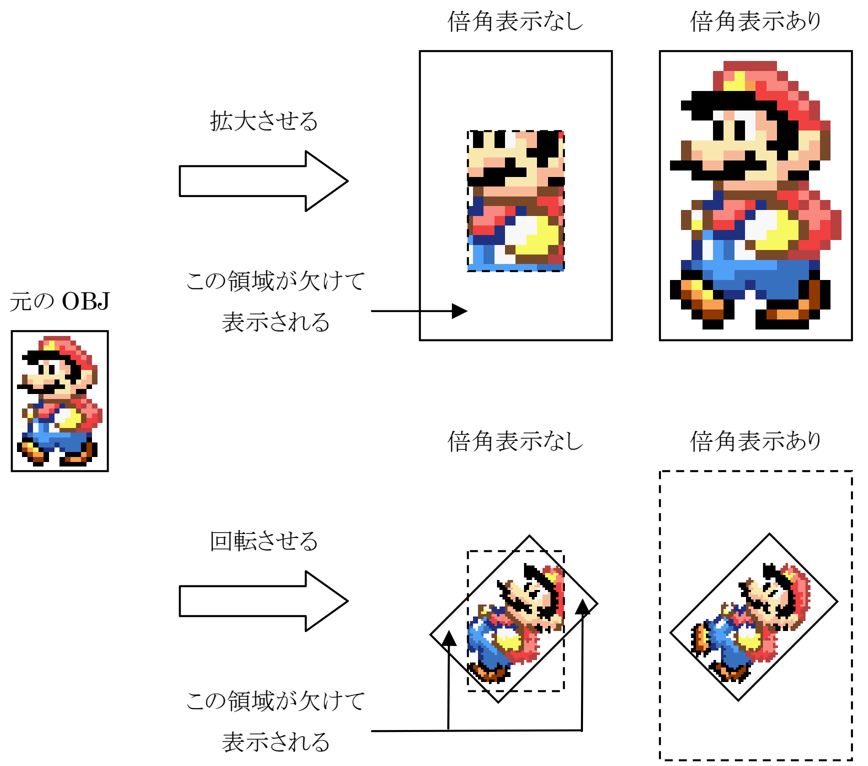


図 3-6 OBJ の倍角表示

○ HVフリップ

水平方向と垂直方向それぞれに反転させて表示する機能です。(図 3-7)



図 3-7 OBJ の HV フリップ

○ モザイク

モザイクをかけて表示する機能です。

かけるモザイクのドット数を指定することができます。（図 3-8）

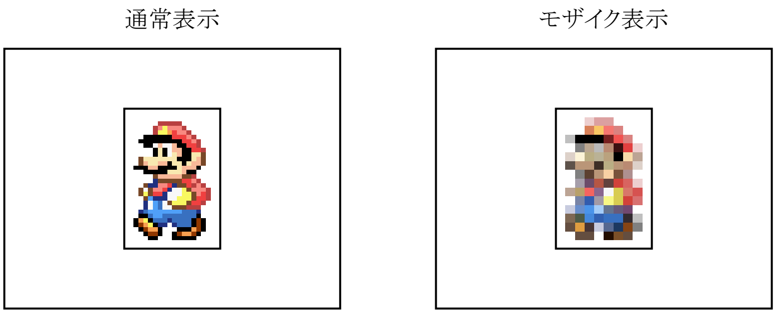


図 3-8 OBJ のモザイク

○ 優先順位

表示上の優先順位を 4 段階で指定します。

優先順位が高い OBJ ほど手前に表示されます。

OBJ ウィンドウに対しては、優先順位を指定することができません。

3.3 BGの機能

BG には、キャラクタ方式とビットマップ方式の 2 種類があります。

- キャラクタ方式　・・・ スクリーンデータを用いてキャラクタデータを参照する方式です。
- ビットマップ方式 ・・・ スクリーンデータやキャラクタデータは用いずに、
  カラー値やカラーインデックス値を直接指定する方式です。

キャラクタ方式、ビットマップ方式を含めて、BGは 表 3-2に示す 5 種類のBGタイプに分けられます。

表 3-2 BG タイプ

<table>
<tr><th>方式</th><th>BG タイプ</th><th>特徴</th></tr>
<tr><td></td><td>3D</td><td>3D グラフィックスエンジンによって生成された画像を表示できます。</td></tr>
<tr><td rowspan="3">キャラクタ方式</td><td>テキスト</td><td>キャラクタ方式の BG です。
唯一 16 色で定義されたキャラクタを扱うことができ、VRAM の消費を抑えることができますが、回転拡大縮小は行えません。</td></tr>
<tr><td>アフィン</td><td>回転拡大縮小を行うことができるキャラクタ方式の BG です。
キャラクタ単位の処理（HV フリップ）を行うことはできません。</td></tr>
<tr><td rowspan="2">アフィン拡張 BG</td><td rowspan="2">以下の 3 種類の中から選択できます。
・ 256 色×16 パレットを使用できるキャラクタ BG
・ 256 色ビットマップ BG
・ 直接カラーを指定できるビットマップ BG</td></tr>
<tr><td rowspan="2">ビットマップ方式</td></tr>
<tr><td>大画面 256 色
ビットマップ</td><td>大画面のビットマップ BG です。
1 面で BG 用に使用する VRAM の最大容量（512K バイト）を使い切るため、他の BG と併用することはできません。3D 面とは併用できます。</td></tr>
</table>

BG は、2D グラフィックスエンジン A、2D グラフィックスエンジン B それぞれに対して BG0～BG3 の 4 面があり、BG0～BG3 をどの BG タイプにするのかは、BG モードによって決まります。

BG モードの種類は 2D グラフィックスエンジン A と 2D グラフィックスエンジン B とで異なっています。

2Dグラフィックスエンジン Aでは、表 3-3に示す 7 種類のBGモードから選択することができます。
2Dグラフィックスエンジン Bでは、表 3-4に示す 6 種類のBGモードから選択することができます。

表 3-3 2D グラフィックスエンジン A の BG モード

<table>
<tr><th>BG モード</th><th>BG0</th><th>BG1</th><th>BG2</th><th>BG3</th></tr>
<tr><td>0</td><td rowspan="6">テキスト／3D
のどちらかを選択</td><td rowspan="6">テキスト</td><td>テキスト</td><td>テキスト</td></tr>
<tr><td>1</td><td>テキスト</td><td>アフィン</td></tr>
<tr><td>2</td><td>アフィン</td><td>アフィン</td></tr>
<tr><td>3</td><td>テキスト</td><td>アフィン拡張 BG</td></tr>
<tr><td>4</td><td>アフィン</td><td>アフィン拡張 BG</td></tr>
<tr><td>5</td><td>アフィン拡張 BG</td><td>アフィン拡張 BG</td></tr>
<tr><td>6</td><td>3D</td><td>×</td><td>大画面 256 色
ビットマップ</td><td>×</td></tr>
</table>

表 3-4 2D グラフィックスエンジン B の BG モード

<table>
<tr><th>BG モード</th><th>BG0</th><th>BG1</th><th>BG2</th><th>BG3</th></tr>
<tr><td>0</td><td rowspan="6">テキスト</td><td rowspan="6">テキスト</td><td>テキスト</td><td>テキスト</td></tr>
<tr><td>1</td><td>テキスト</td><td>アフィン</td></tr>
<tr><td>2</td><td>アフィン</td><td>アフィン</td></tr>
<tr><td>3</td><td>テキスト</td><td>アフィン拡張 BG</td></tr>
<tr><td>4</td><td>アフィン</td><td>アフィン拡張 BG</td></tr>
<tr><td>5</td><td>アフィン拡張 BG</td><td>アフィン拡張 BG</td></tr>
</table>

＊2D グラフィックスエンジン B の BG には、「3D」と「大画面 256 色ビットマップ」の BG タイプは選択できません。

各BG面に設定する項目や適用できる機能は、BGタイプによって異なり、以下の 表 3-5のようになっています。

表 3-5 BG 機能一覧

<table>
<tr><th>方式</th><th></th><th colspan="3">キャラクタ BG</th><th colspan="3">ビットマップ BG</th></tr>
<tr><th rowspan="2">BG タイプ / 仕様・機能</th><th rowspan="2">3D</th><th rowspan="2">テキスト</th><th rowspan="2">アフィン</th><th colspan="3">アフィン拡張 BG</th><th>大画面</th></tr>
<tr><th>256 色×16 パレット</th><th>256 色</th><th>ダイレクトカラー</th><th>256 色ビットマップ</th></tr>
<tr><td>スクリーンサイズ</td><td>256x192</td><td>256x256
512x256
256x512
512x512</td><td>128x128
256x256
512x512
1024x1024</td><td>128x128
256x256
512x512
1024x1024</td><td>128x128
256x256
512x256
256x512</td><td>128x128
256x256
512x256
256x512</td><td>512x1024
1024x512</td></tr>
<tr><td>キャラクタ数</td><td>×</td><td>1024 個</td><td>256 個</td><td>1024 個</td><td>×</td><td>×</td><td>×</td></tr>
<tr><td>色数／パレット</td><td>262144</td><td>16／16
256／1
256／16</td><td>256／1</td><td>256／16</td><td>256／1</td><td>32768</td><td>256／1</td></tr>
<tr><td>回転拡大縮小</td><td>×</td><td>×</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr>
<tr><td>エリアオーバー処理</td><td>×</td><td>×</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr>
<tr><td>HV フリップ</td><td>×</td><td>○</td><td>×</td><td>○</td><td>×</td><td>×</td><td>×</td></tr>
<tr><td>H スクロール</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr>
<tr><td>V スクロール</td><td>×</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr>
<tr><td>モザイク</td><td>×</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr>
<tr><td>優先順位</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr>
</table>

○ スクリーンサイズ

BG のスクリーンデータサイズです。

○ キャラクタ数

8×8ドットのキャラクタを基本キャラクタとした場合に、キャラクタデータ内に定義可能なキャラクタ数です。

○ 色数／パレット

選択できる色数とカラーパレット数です。

カラーパレットを使用しないBGタイプは表示色数を表しています。(表中青色の箇所)

カラーパレットについては「3.4 カラーパレット」で解説します。

○ 回転拡大縮小

BGスクリーンを回転、拡大、縮小して表示する機能です。（図 3-9）

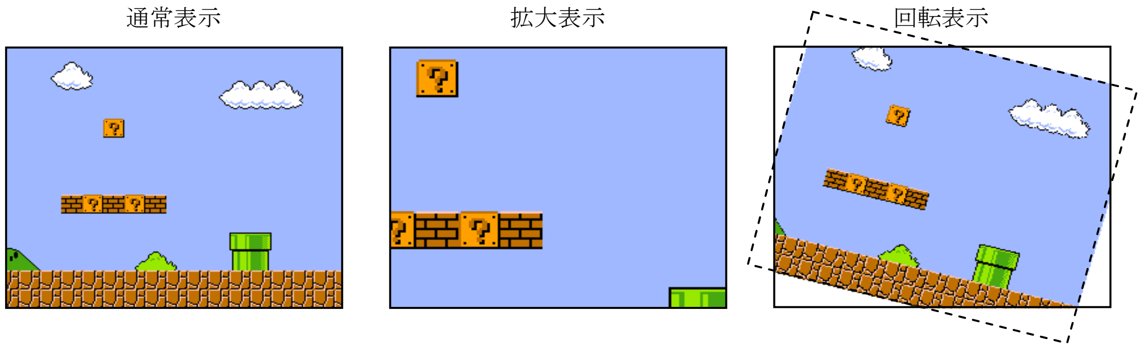


図 3-9 BG の回転拡大縮小

○ エリアオーバー処理

回転拡大縮小機能によって、表示画面がBGスクリーンを超えた場合に、その領域を透明とするか、回りこんで表示するかを選択することができます。（図 3-10）

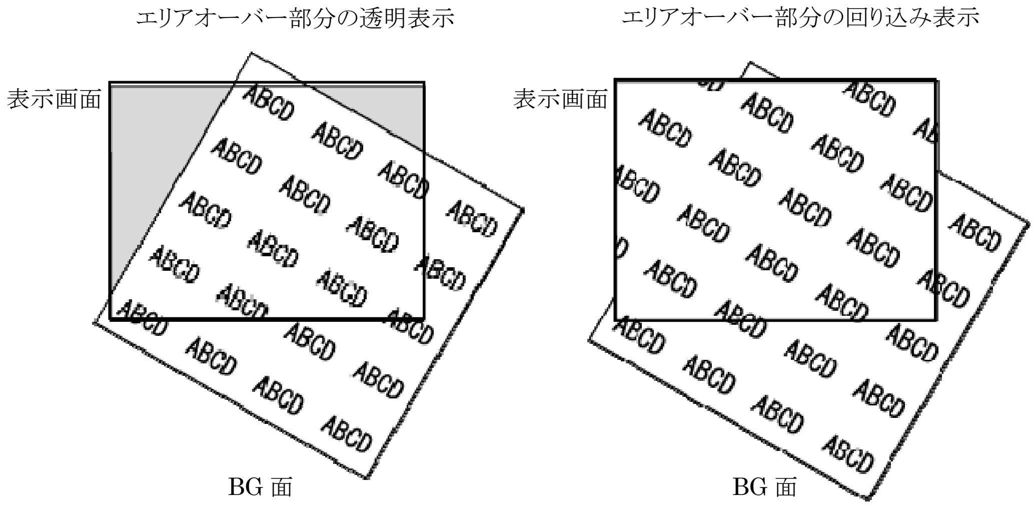


図 3-10 BG のエリアオーバー処理

○ HVフリップ

キャラクタ単位で、水平方向と垂直方向それぞれに反転させて表示する機能です。

○ Hスクロール、Vスクロール

水平方向と垂直方向それぞれにスクロールさせて表示する機能です。

○ モザイク

モザイクをかけて表示する機能です。

かけるモザイクのドット数を指定することができます。（図 3-11）

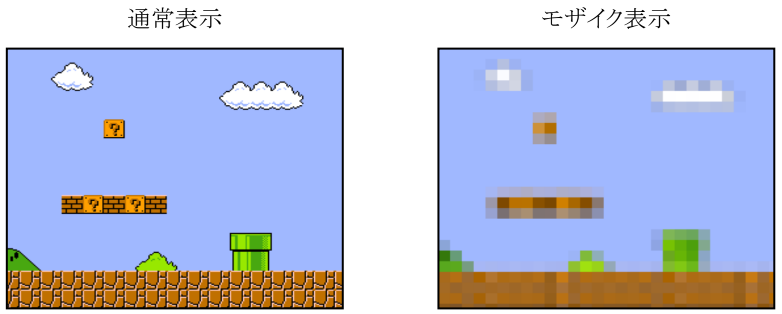


図 3-11 BG のモザイク

○ 優先順位

表示上の優先順位を 4 段階で指定します。

優先順位が高いBG面ほど手前に表示されます。（図 3-12）

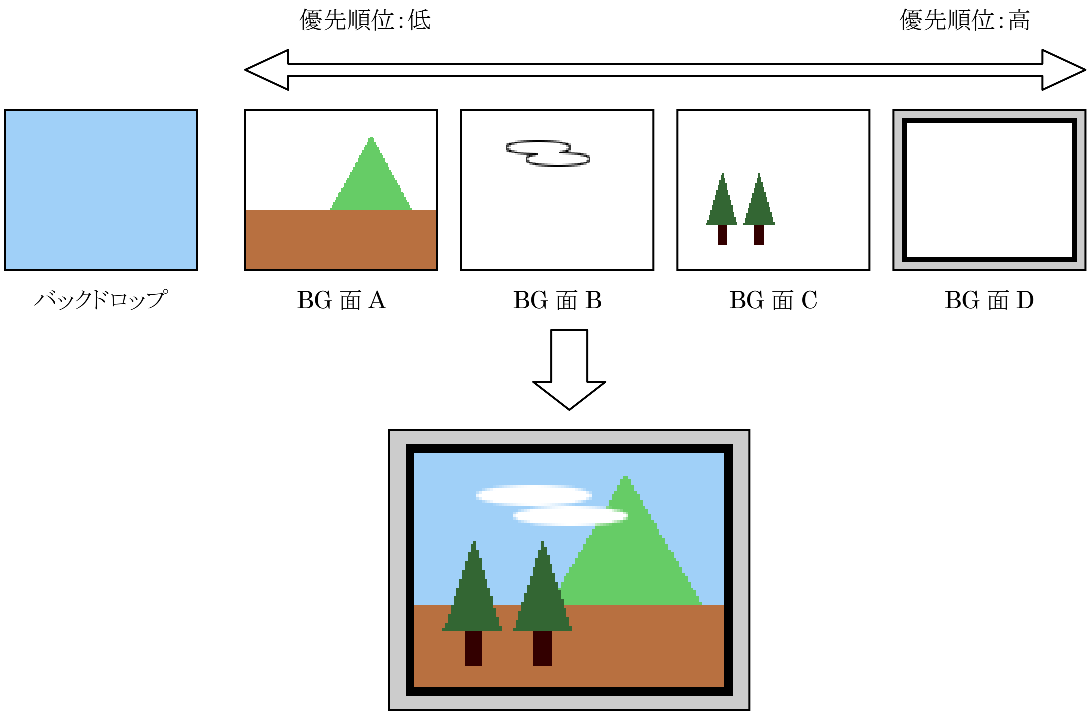


図 3-12 BG の優先順位

3.4 カラーパレット

2 種類のカラーパレットがあります。

- 標準パレット
- 拡張パレット

○ 標準パレット

標準パレットは、プロセッサ内にあるパレット専用のRAMに格納されます。

このRAMは 2Dグラフィックスエンジン A用に 1Kバイト、2Dグラフィックスエンジン B用に 1Kバイトとなっています。

標準パレットは、OBJ用(512 バイト)とBG用(512 バイト)に分けられており、それぞれのパレットに対して、2 種類のパレットタイプから選択して使用します。(図 3-13)

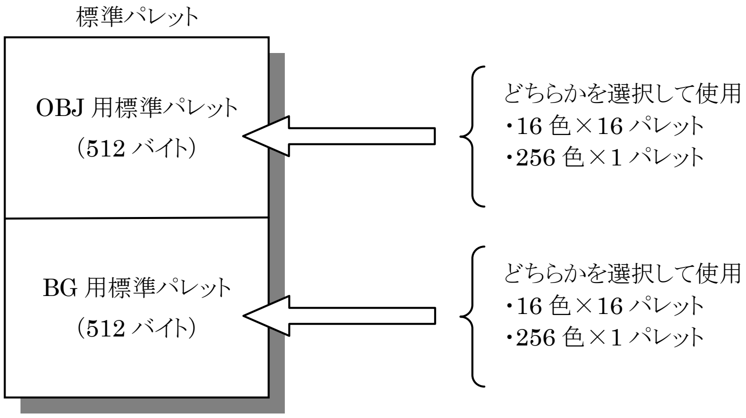


図 3-13 標準パレット

○ 拡張パレット

拡張パレットを使用するためには、VRAMのいくらかの領域を、拡張パレット用に割り当てる必要があります。

割り当てられた拡張パレットは、256 色×16 パレットとして使用します。(図 3-14)

拡張パレットは、256 色×16 パレットモードのOBJ、BGに対してのみ適用することができます。

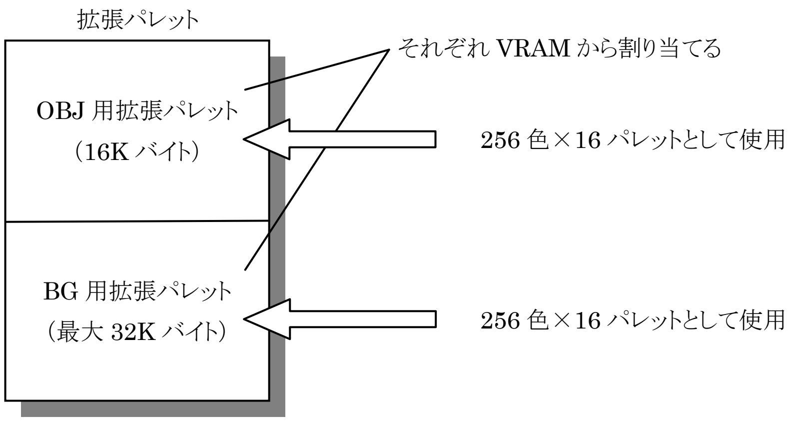


図 3-14 拡張パレット

なお、全てのカラーパレットにおいて、各パレットのインデックス 0 番カラーは透明色として扱われます。

3.5 ウィンドウ

ウィンドウ機能では、以下の3種類のウィンドウを使用することができます。

- ウィンドウ0
- ウィンドウ1
- OBJウィンドウ

○ ウィンドウ0、ウィンドウ1（図 3-15）

矩形領域のウィンドウです。

Hブランク期間中にウィンドウの座標を変更することで、ウィンドウ領域を変形することもできます。

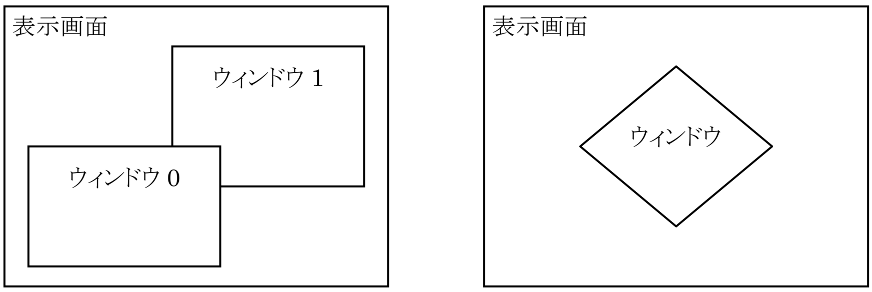


図 3-15 ウィンドウの例

○ OBJウィンドウ

OBJウィンドウに指定されたOBJは、ウィンドウとして使用することができます。

ウィンドウ0，1とは異なり、任意の形状をとることができます。（図 3-16）

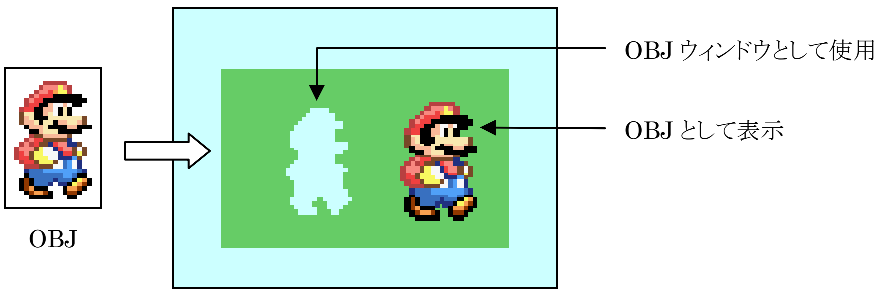


図 3-16 OBJウィンドウ

それぞれのウィンドウの内側、外側に対して、以下の設定をすることができます。

- 各BG面、OBJを表示するかどうか ･･･BGやOBJを表示する領域を限定することができます。
- カラー特殊効果を適用するかどうか ･･･カラー特殊効果を適用する領域を限定することができます。

○ ウィンドウの優先順位

ウィンドウ 0 は常にウィンドウ 1 よりも優先表示され、OBJウィンドウは最も優先順位が低くなります。
ウィンドウの優先順位は変更することができません。（図 3-17）

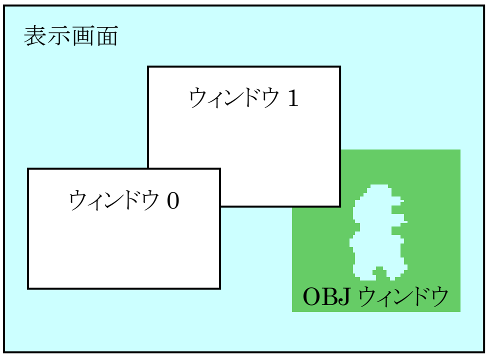


図 3-17 ウィンドウの優先順位

3.6 カラー特殊効果

BG や OBJ、バックドロップ面に対して、以下の特殊効果のいずれかを適用することができます。

- αブレンディング
- 輝度アップ
- 輝度ダウン

○ αブレンディング

指定した 2 面に対して、混ぜ合わせ係数を用いてカラーの混ぜ合わせを行います。

○ 輝度アップ

指定した面の輝度を上げます。

○ 輝度ダウン

指定した面の輝度を下げます。

3.7 AGBとの違い

2D グラフィックス機能について、

『 AGB と何が違うのか？ 』
『 どういった所が改善されたのか？ 』

をまとめますと、主に次のような点が挙げられます。

○ OBJ について

- ビットマップ OBJ が追加された。
- 拡張パレット（256 色×16 パレット）が追加された。
- 同時に保持できるキャラクタ数が増えた。（ただし、データの配置の仕方に制限があります。）
- 1 ラインに今までより多く表示しても消えにくくなった。

○ BG について

- BG タイプが増えた。＝アフィン拡張 BG、大画面 256 色ビットマップ BG が追加された。
- BG モードが増えた。これにより、
  - BG タイプに関わらず、常に BG を 4 面表示することができるようになった。
  - キャラクタ方式の BG とビットマップ方式の BG が混在できるようになった。
- 拡張パレット（256 色×16 パレット）が追加された。

4 3Dグラフィックス

4.1 絵が出来上がるまでの流れ

3Dグラフィックス機能は、GE（ジオメトリエンジン）の処理とRE（レンダリングエンジン）の処理に分けられます。

GEとREの処理を経て、絵が出来上がるまでの大まかな処理の流れを以下に示します。（図 4-1）

① CPU によりコマンド FIFO に書き込まれた描画コマンドが GE に送られます。

② GE は発行された描画コマンドを基に、頂点の計算処理を行います。

③ GE によって作成された頂点の情報は、専用の RAM に格納され、RE に送られます。

④ RE は受け取った頂点の情報を基に、ピクセルの描画処理を行います。

⑤ RE によってレンダリングされた画像は、2D グラフィックスエンジン A の BG0 面上に出来上がります。

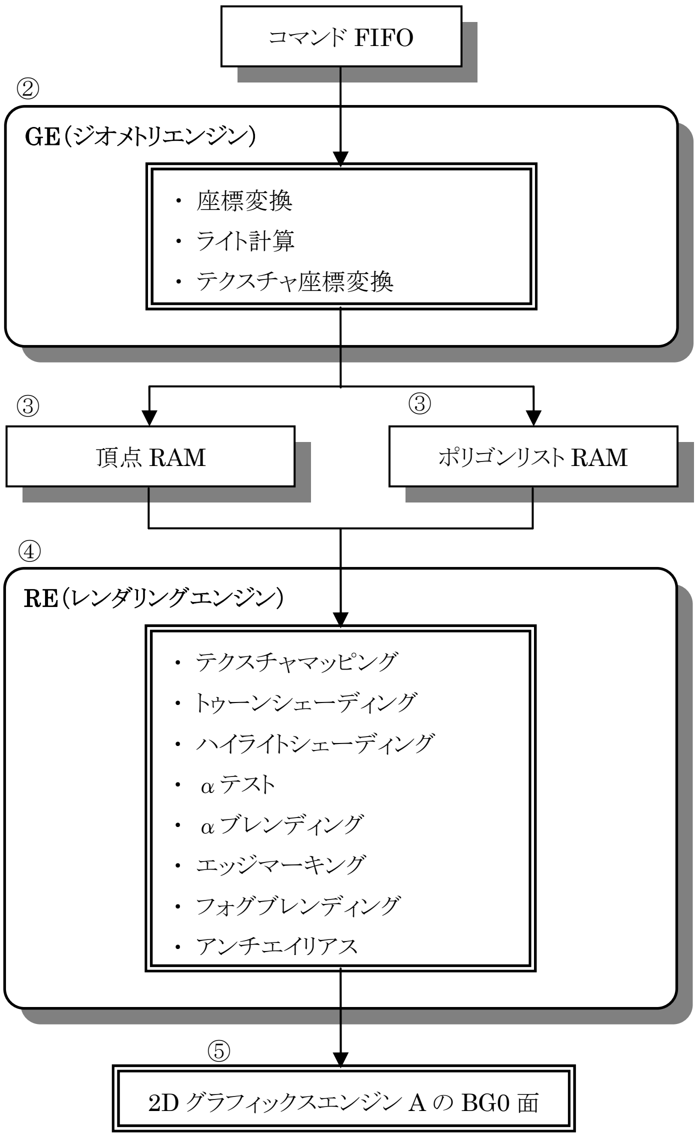


図 4-1 3D グラフィックスで絵が出来上がるまでの流れ

4.2 GE（ジオメトリエンジン）

4.2.1 GEは何をするところか？

GE が行う処理には、次の 3 つがあります。

- 座標変換
- ライト計算
- テクスチャ座標変換

GEは、送られてくる頂点データに対してこれらの処理を行い、出来上がった頂点の情報を、頂点RAM、ポリゴンリストRAMと呼ばれる専用のRAMに格納します。（図 4-2）

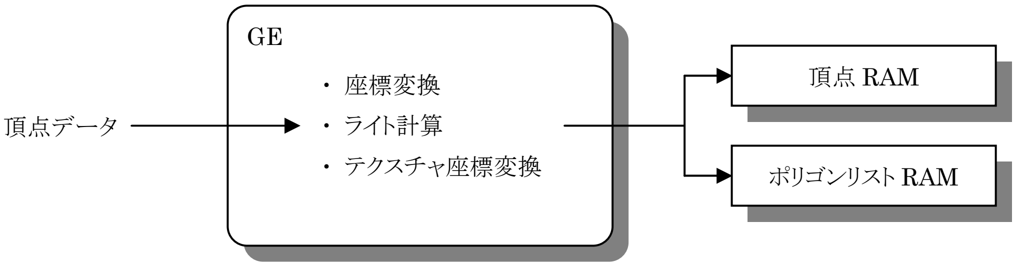


図 4-2 GE が行う処理

頂点 RAM には、処理後の頂点の情報が格納されます。

ポリゴンリストRAMには、各頂点をどのように結んでポリゴンを形成するかという情報が格納されます。（図 4-3）

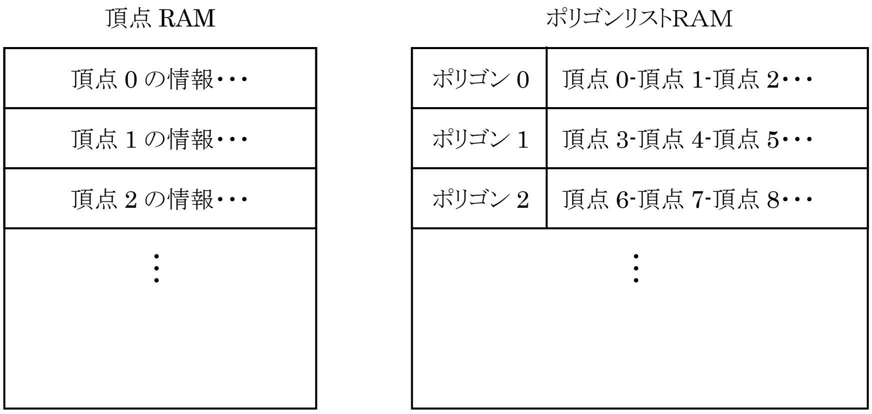


図 4-3 頂点 RAM とポリゴンリスト RAM の概念図

描画できるポリゴンには、次の 4 種類の形状があります。（図 4-4）

- 三角形ポリゴン
- 連結三角形ポリゴン
- 四角形ポリゴン
- 連結四角形ポリゴン

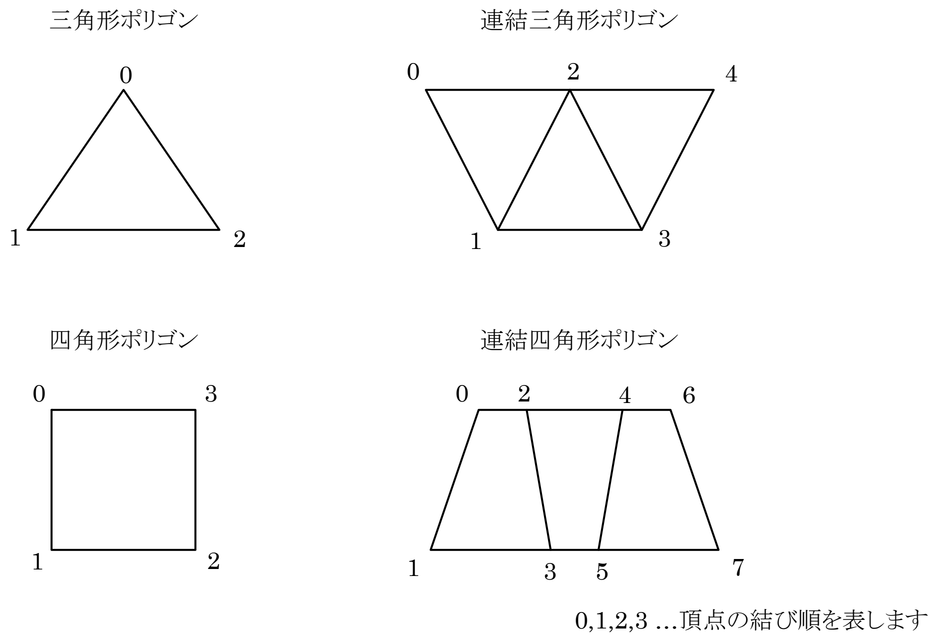


図 4-4 ポリゴンの形状

頂点 RAM に格納することができる頂点数には制限があるため、少ない頂点数でより多くのポリゴンを描画するためには、これらのポリゴンの形状を用いて効率化を図る必要があります。

4.2.2 GEが処理する頂点データ

GE に送ることができる頂点データには以下の 4 種類があります。

- 頂点座標（X, Y, Z）
- 法線ベクトル（Nx, Ny, Nz）
- テクスチャ座標（S, T）
- 頂点カラー（R, G, B）

それぞれの頂点データが送られた時の、GEの処理とデータの流れは、次の図のようになっています。（図 4-5）

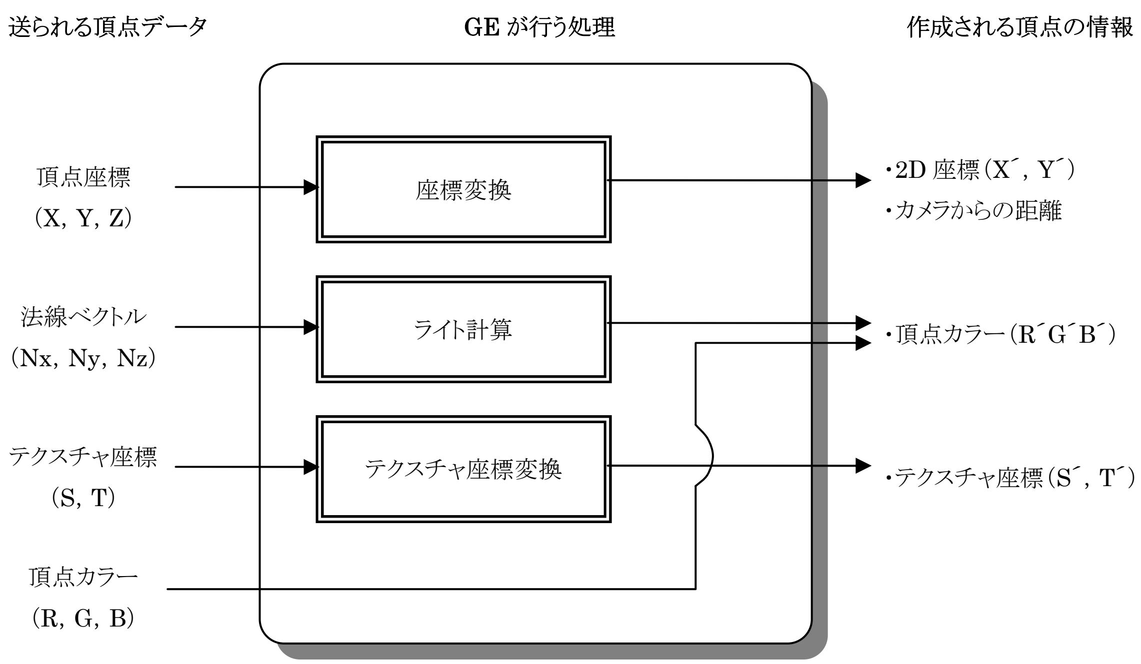


図 4-5 GE に送られる頂点データと処理の流れ

4.2.3 座標変換

GEが行う座標変換には、以下の 2 種類があります。（図 4-6）

- 頂点座標のクリップ座標変換
- 法線ベクトルのベクトル座標変換

○ 頂点座標のクリップ座標変換

頂点座標は、クリップ座標行列による座標変換とクリッピングが行われ、スクリーン上の 2D 座標と、カメラからの距離を表すデプス値に変換されます。（クリップ座標行列の詳細については、ここでは省略します）

○ 法線ベクトルのベクトル座標変換

法線ベクトルは、方向ベクトル行列によりカメラ座標系でのベクトルに変換され、ライト計算に使用されます。（方向ベクトル行列の詳細については、ここでは省略します）

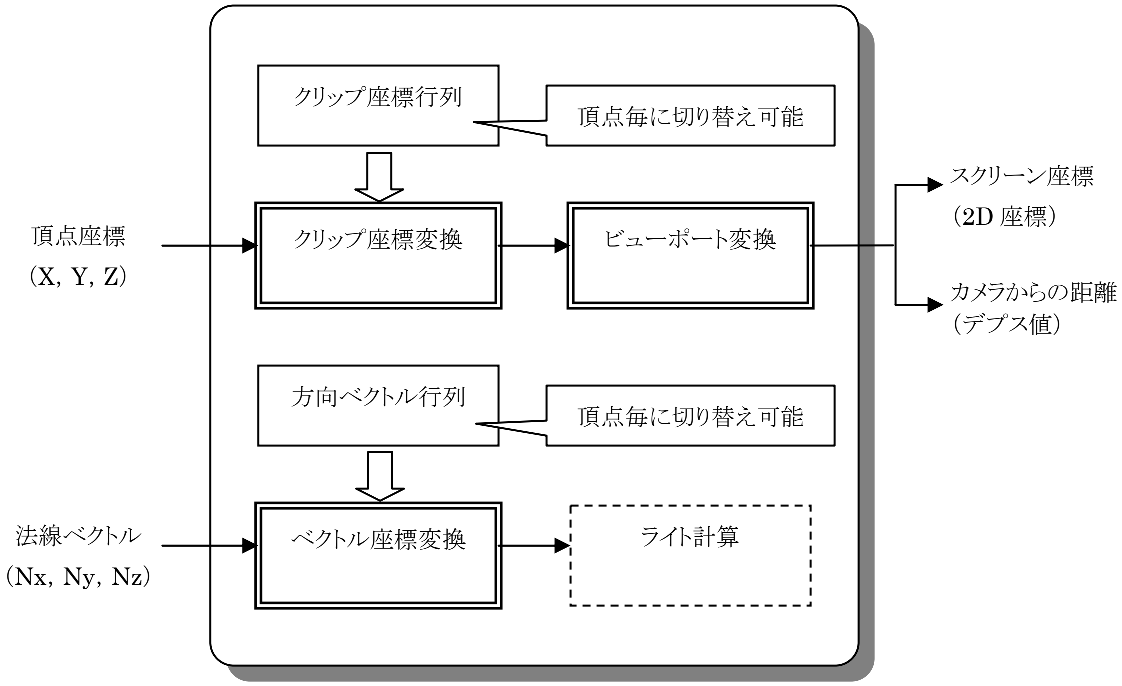


図 4-6 座標変換

クリップ座標行列と方向ベクトル行列は、頂点毎に設定を変更することができます。

これらの変換行列を頂点毎に切り替えることで、ポリゴンの変形（エンベロープやスキニング等）を表現することができます。

4.2.4　ライト計算

ライト計算は、一般的には、光源（＝ライト）が物体の材質（＝ポリゴンのマテリアル）に与える影響を計算し、物体の色や陰影を表現する機能です。物体の材質には、以下に示すような特徴があります。（図 4-7）

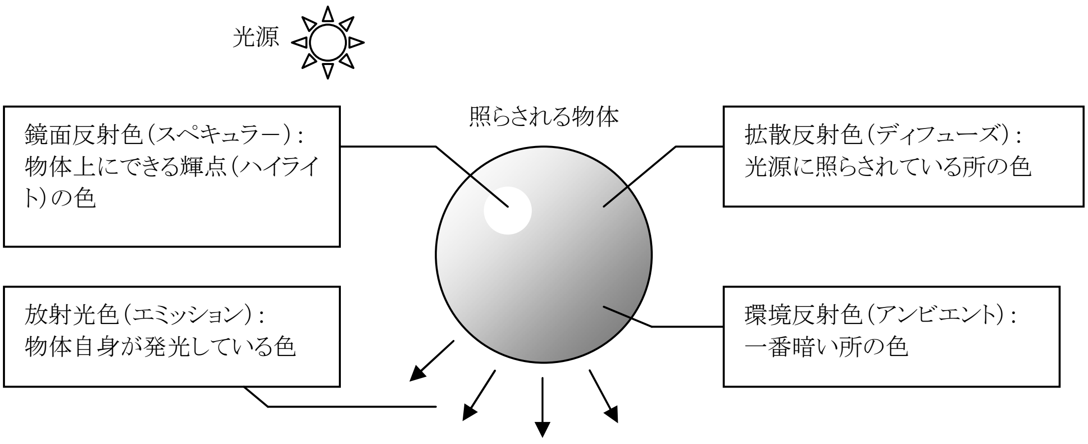


図 4-7 光源と物体の材質

光源として 4 つの平行光源ライト（ライト 0〜ライト 3）をサポートしています。

ライト計算は、この 4 つのライトの設定とポリゴンのマテリアルの設定をもとに実行され、計算結果としてポリゴンの各頂点の頂点カラーが求まります。（図 4-8）

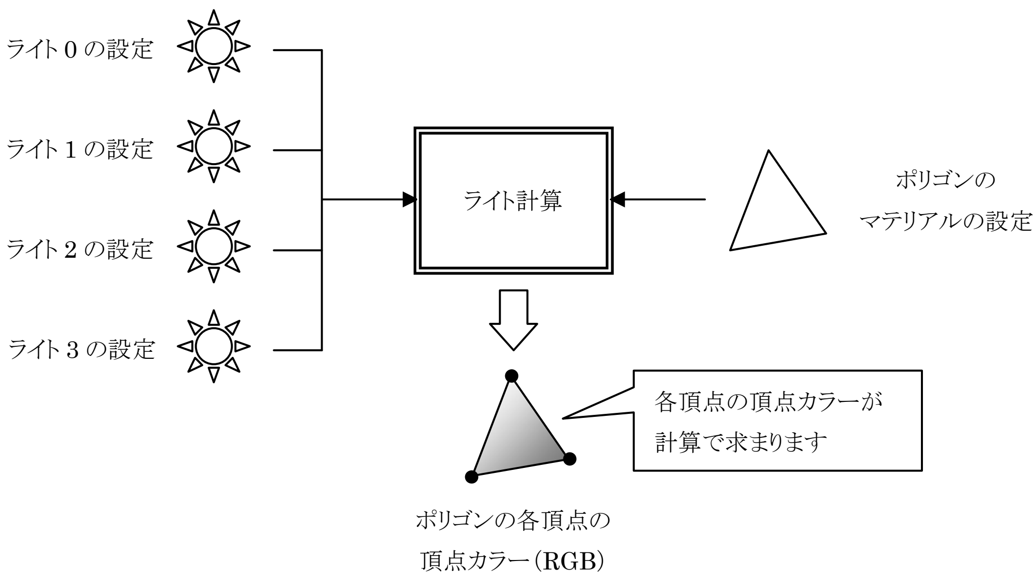


図 4-8 ライト計算

■ ライトの設定

各ライトには以下のパラメータを設定することができます。

- ライトの方向を表すベクトル（Lx, Ly, Lz）
- ライトのカラー（RGB）

■ ポリゴンのマテリアルの設定

ポリゴンのマテリアルには以下のパラメータを設定することができます。

- ディフューズカラー（RGB）：拡散反射色
- アンビエントカラー（RGB）：環境反射色
- スペキュラカラー（RGB）：鏡面反射色
- エミッションカラー（RGB）：放射光色
- どのライトの影響を受けるか（ライト 0～ライト 3 の任意の組み合わせ）
- スペキュラ成分計算時に鏡面反射輝度テーブルを使用する／しない
- 鏡面反射輝度テーブル

以上のことをまとめると、ライトに設定するパラメータとポリゴンのマテリアルに設定するパラメータを用いて、ライト計算は次のようになります。（図 4-9）

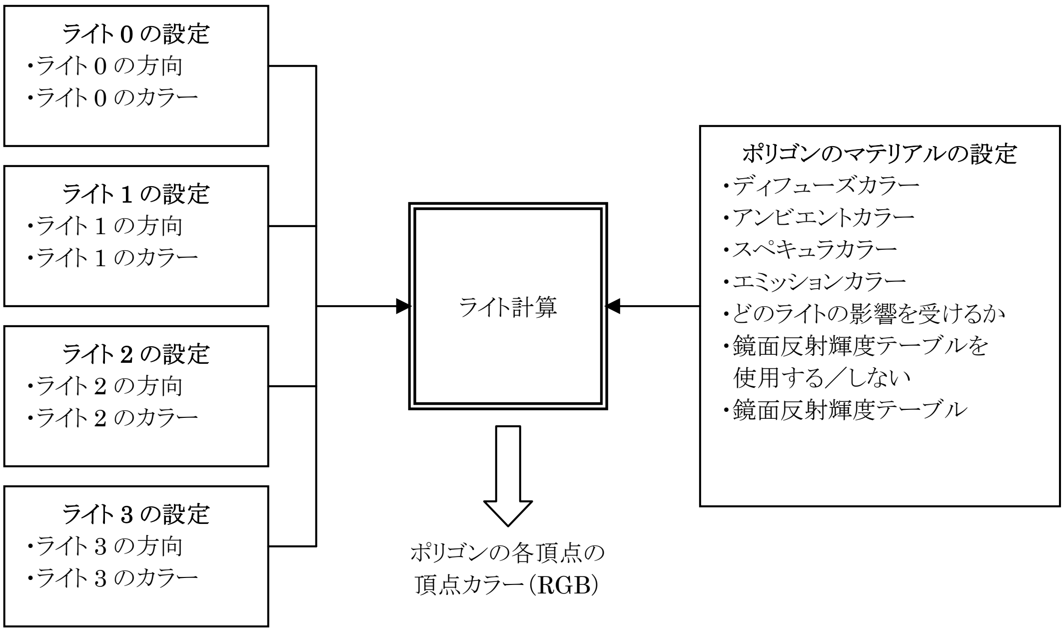
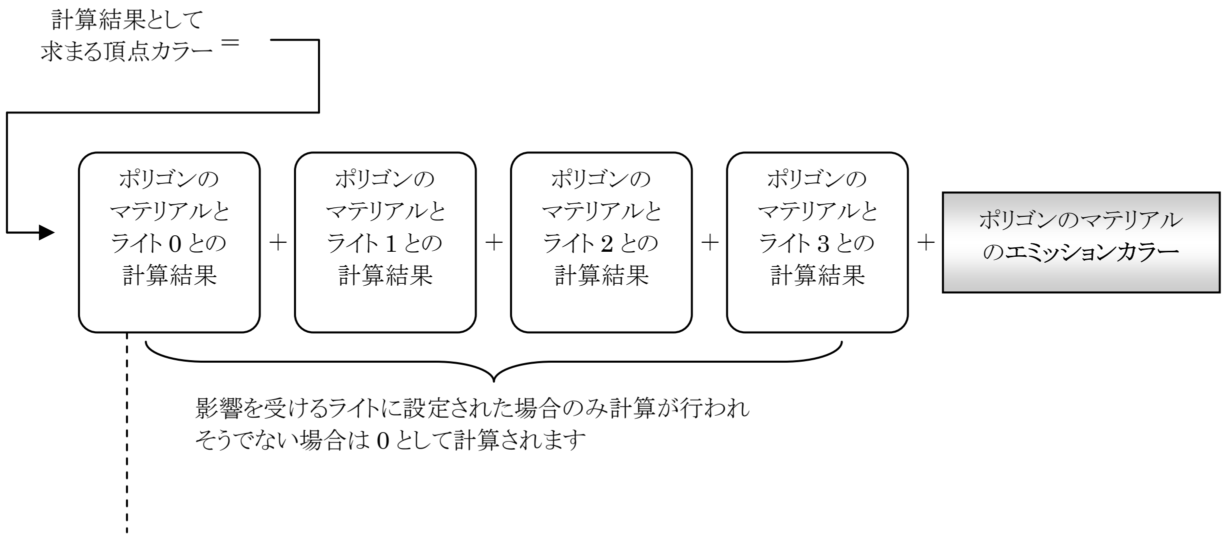


図 4-9 ライト計算に使用するパラメータ

■ ライト計算の計算式

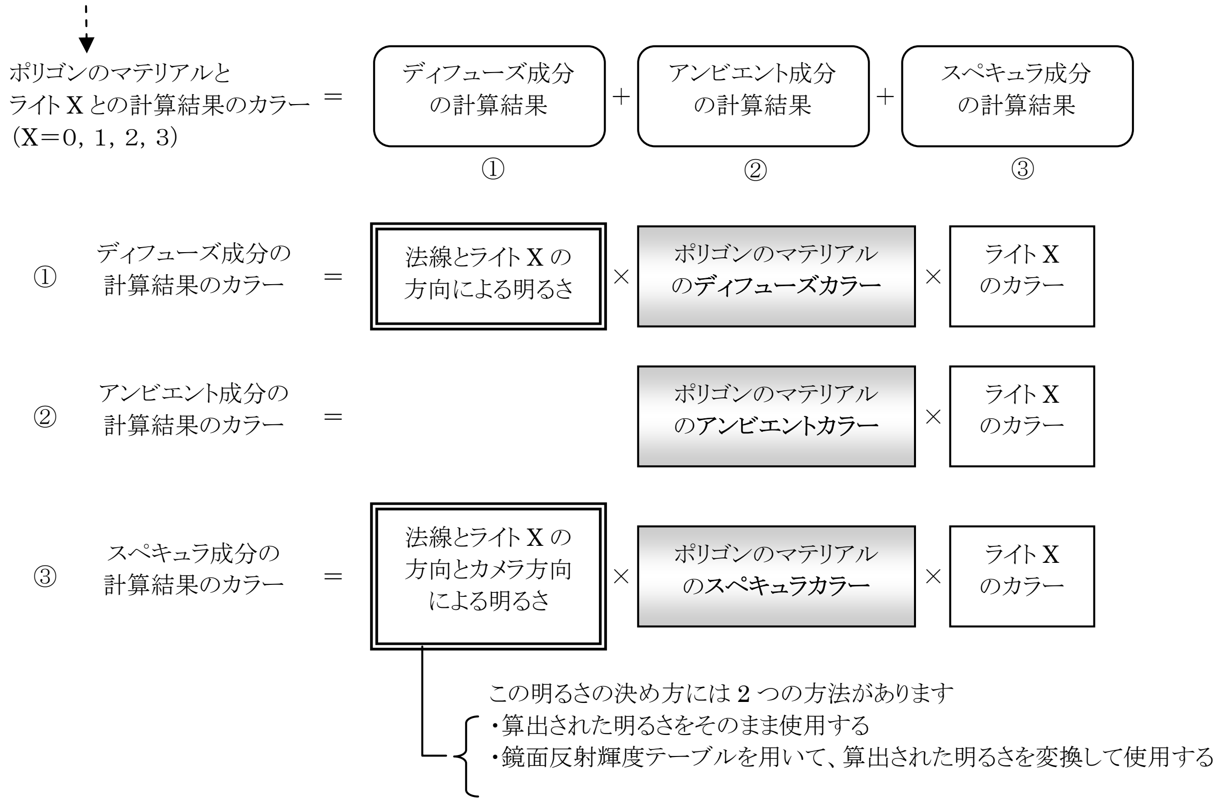
ライト計算は以下の計算式により、RGB 成分毎に行われます。



ポリゴンのマテリアルと各ライトとの計算は、そのライトの影響を受けるように設定した場合のみ行われます。

全てのライトの影響を受けないようにした場合には、エミッションカラーがそのまま結果として求まる頂点カラーとなります。

次に、ポリゴンのマテリアルと各ライトとの計算の詳細を示します。



①、②、③の各成分の計算は必ず行われます。

各成分が必要ない場合には、ポリゴンのマテリアルの該当するカラー（ディフューズカラー、アンビエントカラー、スペキュラカラー）を黒色（R＝G＝B＝0）に設定しておく必要があります。

■ 計算式の数学的な詳細(プログラマ向け)

ライト計算に用いられる計算式の数学的な詳細を以下に示します。(図 4-10)

i はライトの番号を表し、DiffuseCalc(i)、AmbientCalc(i)、SpecularCalc(i)は、ライトi の影響を受ける場合のみ計算されます。また、計算式中で用いる各ベクトルは単位ベクトルとします。

$$\text{計算結果として求まる頂点カラー} = \sum_{i=0}^{3} \left(\text{DiffuseCalc}(i) + \text{AmbientCalc}(i) + \text{SpecularCalc}(i) \right) + \boxed{\text{EmissionColor}}$$

DiffuseCalc(i):

$$= \max[\,0,\ -\vec{L}(i)\cdot\vec{N}\,] \times \boxed{\text{DiffuseColor}} \times \boxed{\text{LightColor}(i)}$$

AmbientCalc(i):

$$= \boxed{\text{AmbientColor}} \times \boxed{\text{LightColor}(i)}$$

SpecularCalc(i):

$$= \text{Shininess}(i) \times \boxed{\text{SpecularColor}} \times \boxed{\text{LightColor}(i)}$$

Shininess(i):

鏡面反射輝度テーブルを使用しない $= \max[\,0, \cos 2\theta\,]$

鏡面反射輝度テーブルを使用する $= \text{ShininessTable}[\,\max[\,0, \cos 2\theta\,]\,]$

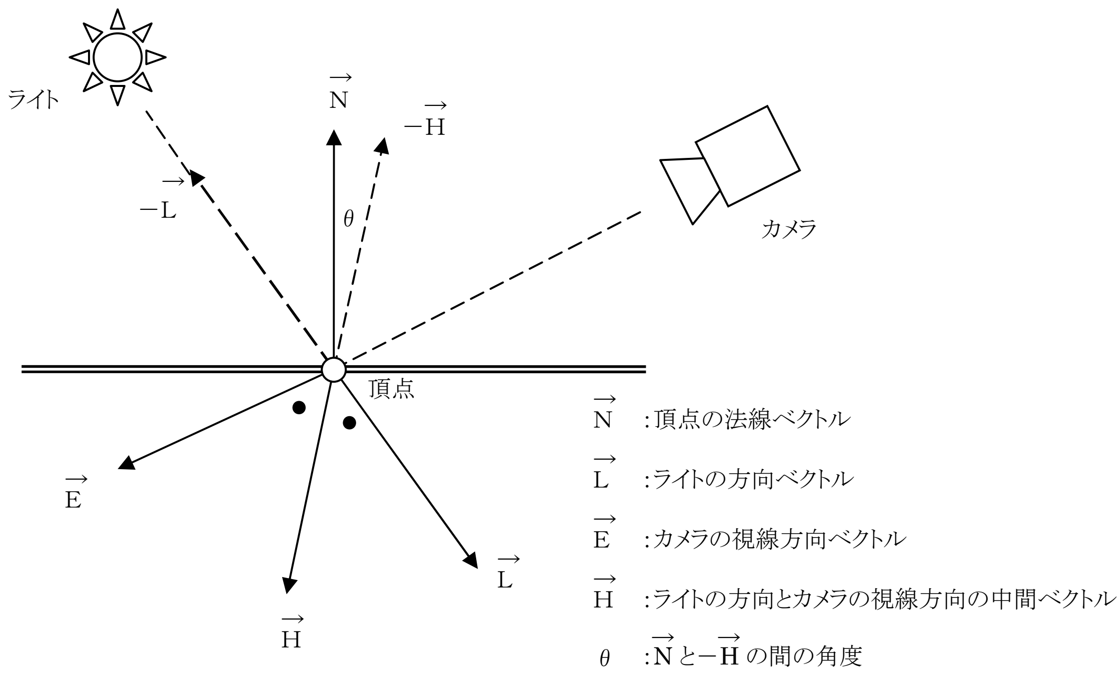


図 4-10 ライト計算式の数学的な詳細

■ 鏡面反射輝度テーブル

鏡面反射輝度テーブルは、スペキュラ成分計算で用いる鏡面反射輝度を変換するためのテーブルです。

鏡面反射輝度テーブルにより、スペキュラハイライトの明るさを制御することができます。

4.2.5 テクスチャ座標変換

テクスチャ座標変換は、ポリゴンにテクスチャを貼るためのテクスチャ座標を生成します。

テクスチャ座標変換の流れは次のようになります。(図 4-11)

① 入力ソースを以下の 3 種類の頂点データから指定します。

- 頂点座標(X, Y, Z)
- 法線ベクトル(Nx, Ny, Nz)
- テクスチャ座標(S, T)

② テクスチャ座標変換用の行列(テクスチャ行列)を設定します。

③ ①, ②で指定した入力ソースとテクスチャ行列が行列計算により掛け合わされ、テクスチャ座標が生成されます。

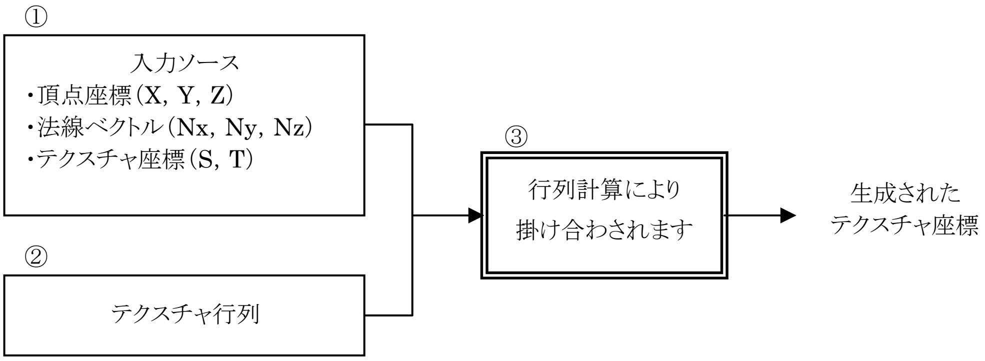


図 4-11 テクスチャ座標変換

入力ソースとテクスチャ行列の組み合わせにより、様々なテクスチャの貼り方が表現できます。

以下に、代表的な具体例を示します。

■ テクスチャの回転拡大縮小(図 4-12、図 4-13)

入力ソース　: テクスチャ座標(S, T)
テクスチャ行列 : 回転行列

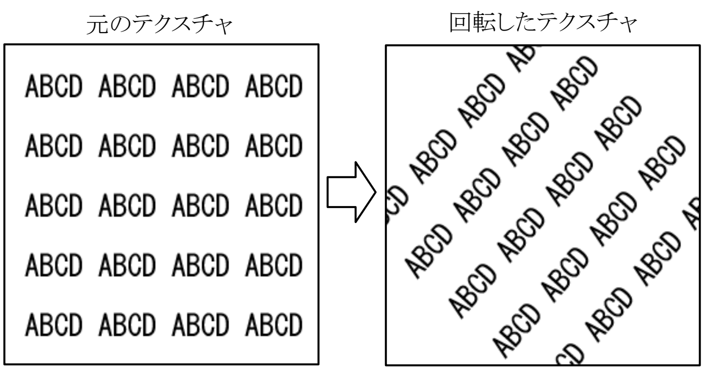


図 4-12 テクスチャの回転

入力ソース　　　：テクスチャ座標(S, T)
テクスチャ行列　：拡大行列

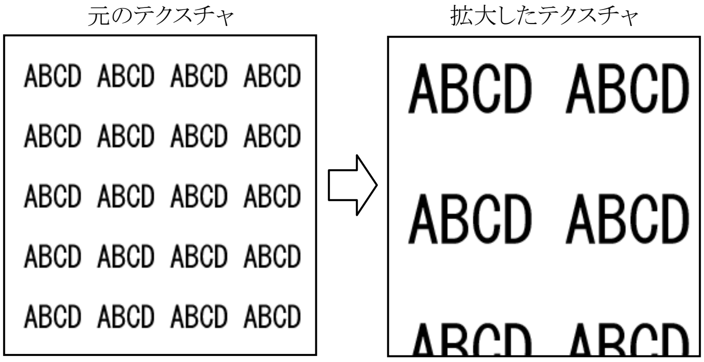


図 4-13 テクスチャの拡大

■　テクスチャの環境マッピング(図 4-14)

入力ソース　　　：法線ベクトル(Nx, Ny, Nz)
テクスチャ行列　：方向ベクトル行列

周囲の環境の写りこみを表現することができます。



＊図は環境マッピングのイメージ画像であり、
実際の表示とは異なる可能性があります。

図 4-14 テクスチャの環境マッピング

4.2.6 ポリゴンの属性

GE の計算処理で直接扱われるものではありませんが、ポリゴン毎に以下の属性値を設定することができます。

なお、これらの属性値は RE の描画処理で使用されます。

- ポリゴン ID
- ポリゴンの α 値
- ポリゴンモード
- フォグをかける／かけない

○ ポリゴン ID

ポリゴン ID には 0～63 の任意の値（6 ビット値）を設定することができます。

○ ポリゴンの α 値

ポリゴンの α 値は、ポリゴンの透明度を表し 0～31 の任意の値（5 ビット値）を設定することができます。

設定された値によって、以下のような扱われ方の違いがあります。

ポリゴンの α 値
- 0 ・・・ワイヤフレームポリゴンとして描画されます。
- 1～30 ・・・半透明ポリゴンとして扱われます。
- 31 ・・・不透明ポリゴンとして扱われます。

○ ポリゴンモード

ポリゴンモードは、REでポリゴンカラーやテクスチャカラーの混ぜ合わせをする際のモードのことです。

以下の 4 種類から選択することができます。

各モードの詳細については、「4.3.4 ポリゴンカラーとテクスチャカラーの混ぜ合わせ」で解説します。

ポリゴンモード
- デカルモード
- モジュレーションモード
- トゥーンシェーディング
- ハイライトシェーディング

○ フォグをかける／かけない

フォグをかけるかどうかを選択することができます。

フォグの詳細については、「4.3.8 フォグブレンディング」で解説します。

4.2.7　補足 1：ポリゴンのカラーの決め方

ポリゴンのカラーの決め方は、大きく言って以下に示す 4 通りの方法に分けられます。（図 4-15）

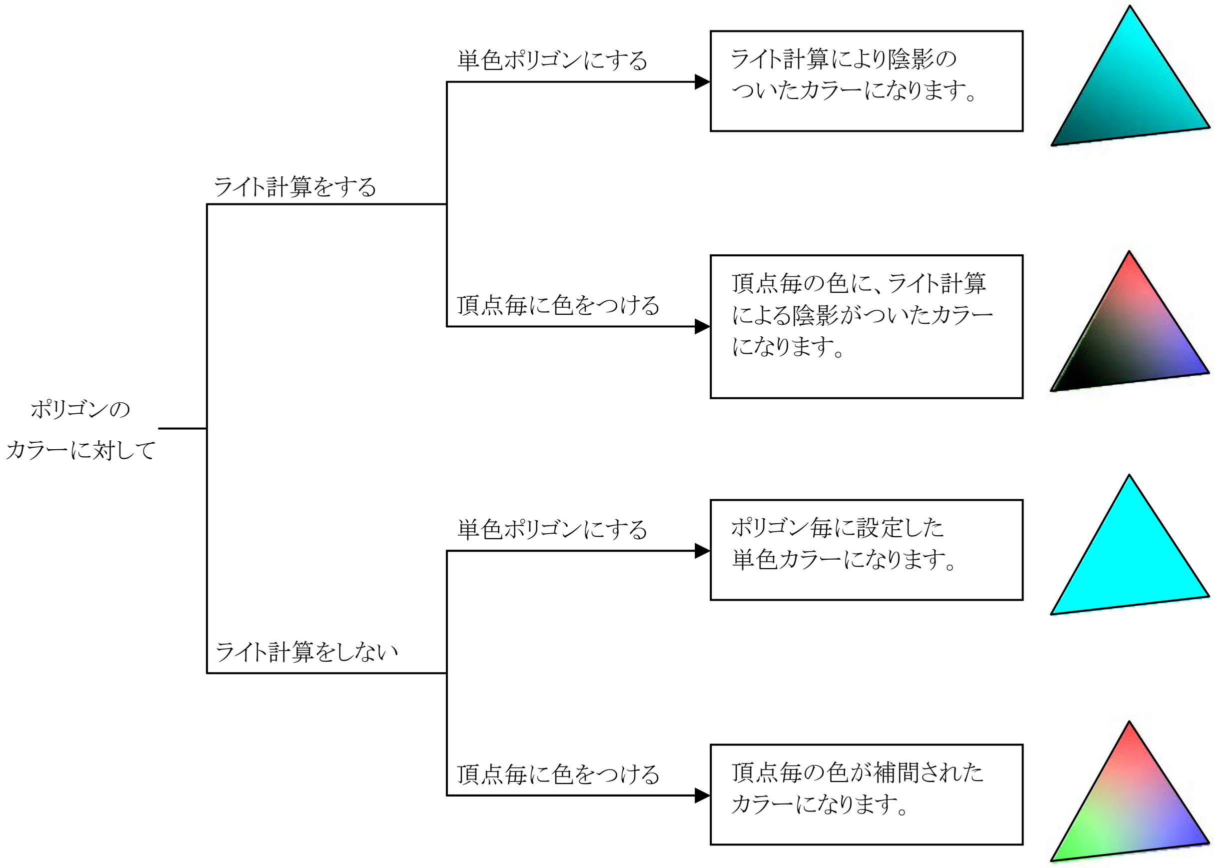


図 4-15 ポリゴンのカラーの決め方

＊上記の内容は、開発機で実際に試した結果に基づいています。

> 注意事項：
> 上記の 4 通りの中で、上から 2 つめの『ライト計算をし、頂点毎に色をつける』については、ポリゴンの頂点の法線が同じであると、GE に同じコマンド（Normal コマンド）を複数回送らなければならず、処理の増加につながるため、実現に際しては注意が必要です。

4.3 RE（レンダリングエンジン）

4.3.1 REは何をするところか？

RE は、GE によって作成された結果をもとに、ポリゴンの描画を行います。

ポリゴンの描画を実行する際には、以下に示す様々な描画機能を適用することができます。（図 4-16）

- テクスチャマッピング
- ポリゴンの RGBA とテクスチャの RGBA の混ぜ合わせ
  - デカルモード
  - モジュレーションモード
  - トゥーンシェーディング
  - ハイライトシェーディング
- αテスト
- αブレンディング
- エッジマーキング
- フォグブレンディング
- アンチエイリアス

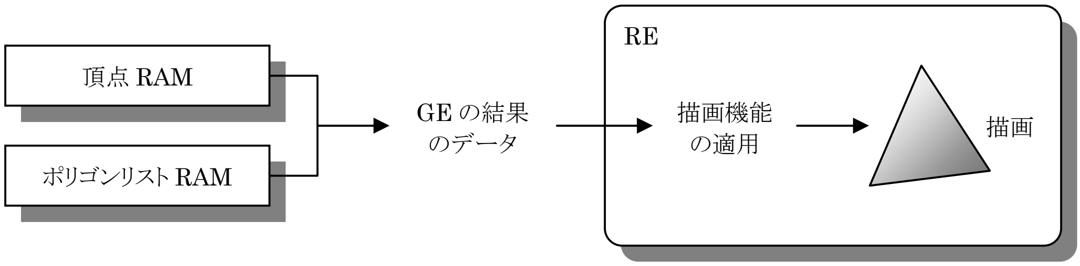


図 4-16 RE が行う処理

4.3.2 ラインバッファ方式とフレームバッファ方式

RE が行うレンダリング方式は、一般的なレンダリング方式であるフレームバッファ方式とは異なり、画面上部から 1 ラインずつ描き込みを行い表示する、ラインバッファ方式と呼ばれる方式で行われます。

○ 一般的なフレームバッファ方式

フレームバッファへの全ての描画が終了した後に表示用に転送されます。(図 4-17)

1 描画フレーム内で描画が終了しない場合は、処理落ち(同じ画像が連続して表示される)が発生します。

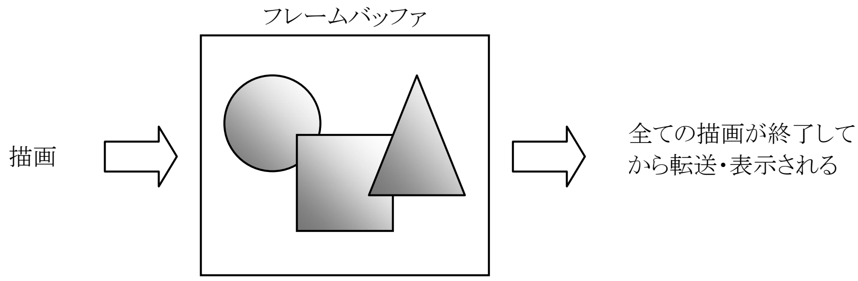


図 4-17 一般的なフレームバッファ方式

○ ラインバッファ方式

1 ライン分の画像データを格納するカラーバッファに上部ラインから順に描画を行い、描画の終わったラインから順に転送・表示します。(図 4-18)

REはカラーバッファを 48 ライン分保持しており、表示前のラインデータを描き貯めておくことができます。

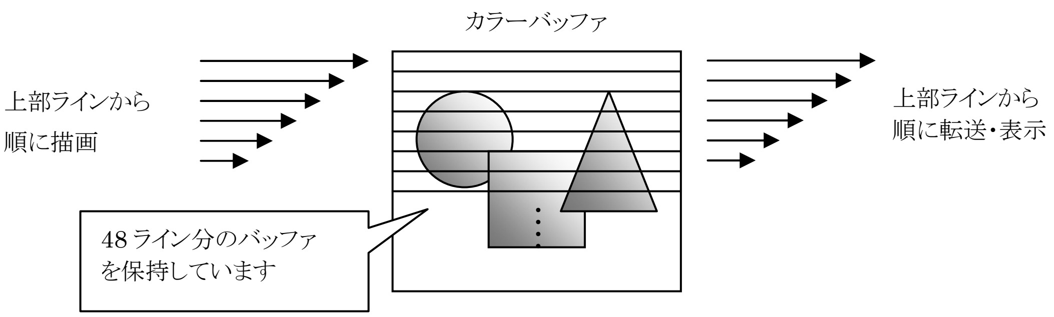


図 4-18 ラインバッファ方式

1 ラインに描画するポリゴン数やピクセル数が RE の処理能力よりも多くなると、レンダリングが表示に間に合わなくなり、表示される画像がくずれます。

4.3.3 テクスチャマッピング

テクスチャ毎に以下の設定ができます。

- サイズ（幅×高さ）
- フォーマットとパレットカラー0 の扱われ方
- 繰り返し方

○ サイズ（幅×高さ）

テクスチャの幅と高さは、それぞれ以下の 8 種類のサイズから選択します。（図 4-19）
なお、幅と高さが同じである必要はなく、長方形サイズにすることもできます。

・ 8　　・ 128
・ 16　　・ 256
・ 32　　・ 512
・ 64　　・ 1024

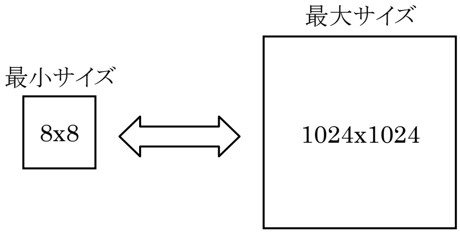


図 4-19 テクスチャサイズ

○ フォーマットとパレットカラー0 の扱われ方

テクスチャのフォーマットには以下の 7 種類があります。

- 4 色パレットテクスチャ
- 16 色パレットテクスチャ
- 256 色パレットテクスチャ
- 4×4 テクセル圧縮テクスチャ
- A5I3 半透明テクスチャ
- A3I5 半透明テクスチャ
- ダイレクトカラーテクスチャ

ダイレクトカラーテクスチャ以外の全てのフォーマットは、カラーパレットを使用するカラーインデックス方式のフォーマットです。カラーパレット上の 1 色のフォーマットは RGB555 となっており、α 成分はありません。

■ 4 色、16 色、256 色パレットテクスチャ

それぞれ、4 色、16 色、256 色から選択可能なカラーインデックスフォーマットです。
これら 4 つのフォーマットに対しては、パレットカラー0（カラーパレット中のインデックス 0 番のカラー）の扱われ方を以下の 2 種類から選択することができます。（図 4-20）

- パレットカラー0 を通常色として扱う
- パレットカラー0 を透明色（抜き）として扱う

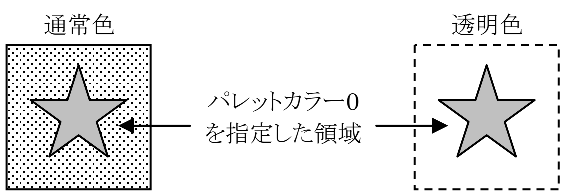


図 4-20 パレットカラー0 の扱われ方

■ 4×4 テクセル圧縮テクスチャ

4×4 テクセル毎にデータを圧縮して持つカラーインデックスフォーマットです。
圧縮アルゴリズムの詳細については、ここでは省略します。(TWL プログラミングマニュアルを参照してください)

■ A5I3 半透明テクスチャ

カラーは 8 色(3 ビット)のパレットから選択し、アルファはパレットを使用せずに 5 ビットのアルファ値を直接指定するフォーマットです。5 ビットのアルファ値により、32 段階の半透明を表現することができます。

■ A3I5 半透明テクスチャ

カラーは 32 色(5 ビット)のパレットから選択し、アルファはパレットを使用せずに 3 ビットのアルファ値を直接指定するフォーマットです。3 ビットのアルファ値により、8 段階の半透明を表現することができます。

■ ダイレクトカラーテクスチャ

カラーインデックス方式でない唯一のフォーマットで、RGBA5551(16 ビット)で色を指定します。
1 ビットのアルファ値を持っているため、抜きの表現が可能です。

各フォーマットの比較を以下に示します。(表 4-1)

表 4-1 テクスチャフォーマットの比較

フォーマット	選択可能な色数	1 テクセル当たりのビット数	抜きの表現	半透明の表現
4 色パレットテクスチャ	4	2	○*	×
16 色パレットテクスチャ	16	4	○*	×
256 色パレットテクスチャ	256	8	○*	×
4×4 テクセル圧縮テクスチャ	4 (4×4 テクセル毎)	3	○	×
A5I3 半透明テクスチャ	8	8 (カラー:3 ビット アルファ:5 ビット)	○	○ (32 段階)
A3I5 半透明テクスチャ	32	8 (カラー:5 ビット アルファ:3 ビット)	○	○ (8 段階)
ダイレクトカラーテクスチャ	32768	16 (RGBA5551)	○	×

* :パレットカラー0 を透明色として設定した場合

○　繰り返し方

テクスチャ座標がテクスチャサイズを超えた領域での、テクスチャの繰り返し方です。
S, T方向それぞれに対して、以下の 3 種類から選択できます。（図 4-21）

- 何もしない　・・・テクスチャの端の色が伸ばされます（いわゆるクランプです）。
- リピート　・・・テクスチャが繰り返されます。
- フリップ　・・・テクスチャが反転しながら繰り返されます（いわゆるミラーです）。

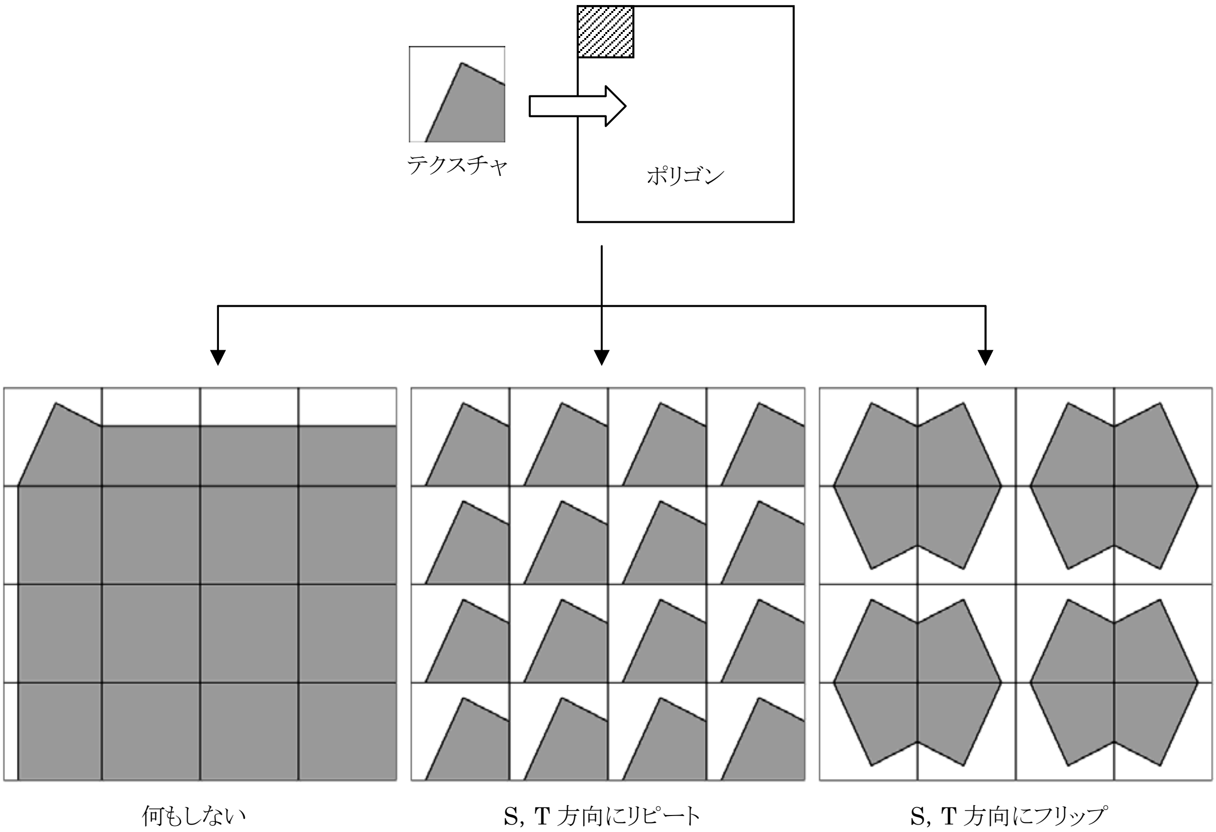


図 4-21 テクスチャの繰り返し方

4.3.4 ポリゴンカラーとテクスチャカラーの混ぜ合わせ

ポリゴン内部の各ピクセルのRGBAは、GEにより算出された各頂点カラー(RGB)とポリゴンのα値から算出されます。(図 4-22)

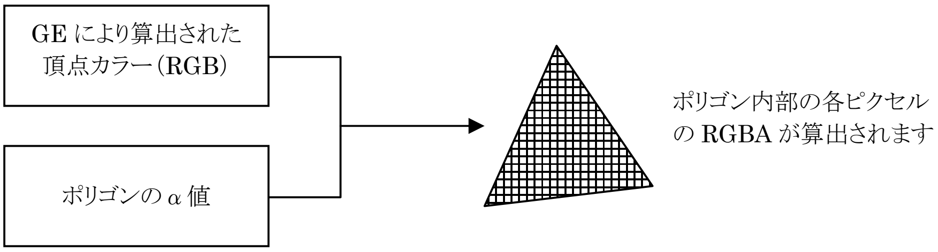


図 4-22 ポリゴン内部の各ピクセルの RGBA

算出された各ピクセルの RGBA は、テクスチャの RGBA と混ぜ合わされます。

混ぜ合わせ方には、ポリゴンに設定されたポリゴンモードによって以下の 4 種類があります。

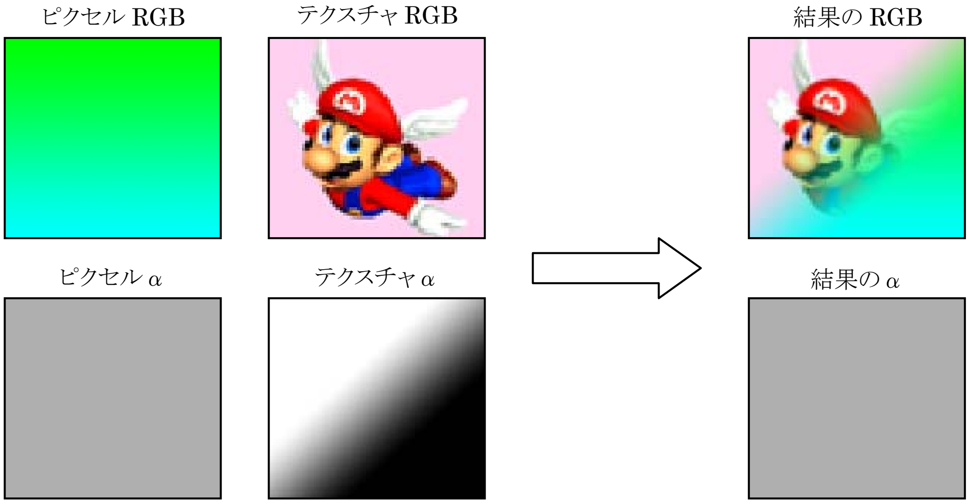
- デカルモード
- モジュレーションモード
- トゥーンシェーディング
- ハイライトシェーディング

○ デカルモード

デカルモードは、テクスチャのα値を混ぜ合わせ比率にして、各ピクセルのRGBとテクスチャのRGBを混ぜ合わせます。α成分に関しては、各ピクセルのα値がそのまま出力されます。(図 4-23)

混ぜ合わせた結果の RGB	＝	各ピクセルの RGB とテクスチャの RGB とを テクスチャのα値を使用して混ぜ合わせた RGB
混ぜ合わせた結果のα値	＝	各ピクセルのα値

＜A5I3 半透明テクスチャ、A3I5 半透明テクスチャの場合＞



＜A5I3 半透明テクスチャ、A3I5 半透明テクスチャでない場合＞

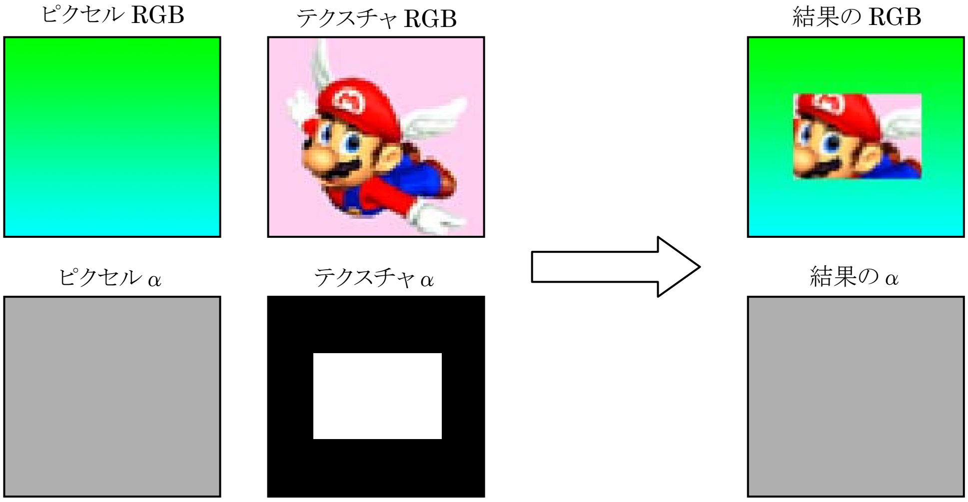


図 4-23 デカルモード

○ モジュレーションモード

モジュレーションモードは、各ピクセルのRGBAとテクスチャのRGBAを掛け合わせて混ぜ合わせます。（図4-24）

混ぜ合わせた結果の RGB	＝	各ピクセルの RGB	×	テクスチャの RGB
混ぜ合わせた結果の α 値	＝	各ピクセルの α 値	×	テクスチャの α 値

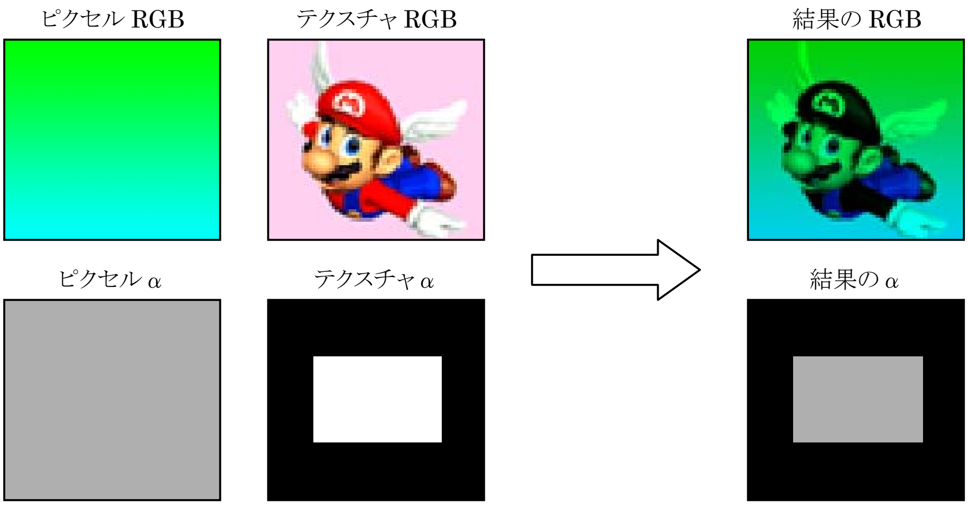


図 4-24 モジュレーションモード

○ トゥーンシェーディング

トゥーンシェーディングでは、トゥーンテーブルと呼ばれるRGB値変換用のテーブルを使用します。

トゥーンテーブルには、任意の 32 種類のカラー(RGB)を設定することができ、各ピクセルのR成分を元にしてRGB値の変換が行われます。(図 4-25)

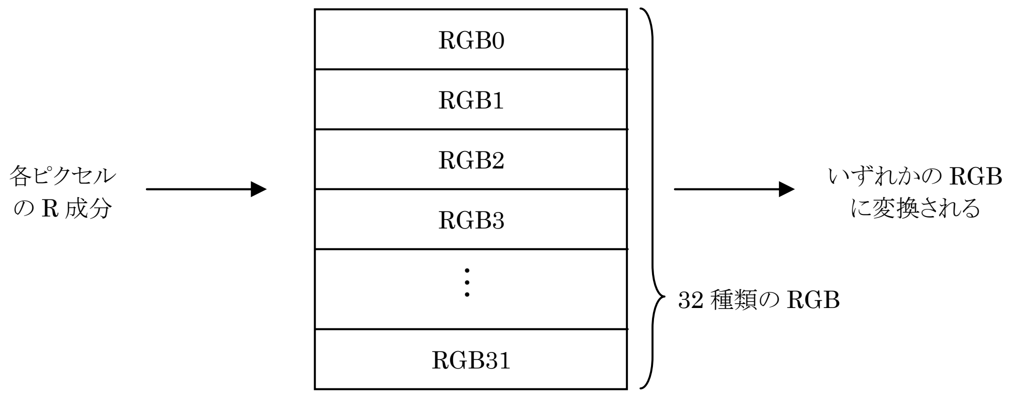


図 4-25 トゥーンテーブル

トゥーンシェーディングは、トゥーンテーブルによって変換されたRGBを新しいピクセルのRGBとして、テクスチャのRGBAとの混ぜ合わせを行います。なお、テクスチャのRGBAとの混ぜ合わせには、前述のモジュレーションモードが用いられます。(図 4-26)

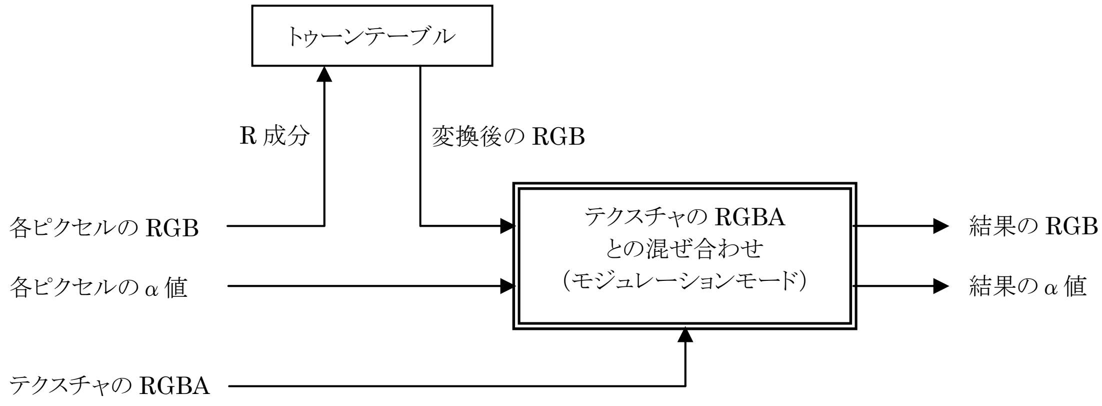


図 4-26 トゥーンシェーディング

○　ハイライトシェーディング

ハイライトシェーディングも、トゥーンシェーディングと同様にトゥーンテーブルによるRGBの変換が行われます。

ハイライトシェーディングでは、まず、各ピクセルのR成分を新しいピクセルのRGBとして、テクスチャのRGBAとの混ぜ合わせを行います。次に、混ぜ合わされた結果のRGBに、トゥーンテーブルによって変換されたRGBが加算されます。

なお、テクスチャのRGBAとの混ぜ合わせには、前述のモジュレーションモードが用いられます。（図 4-27）

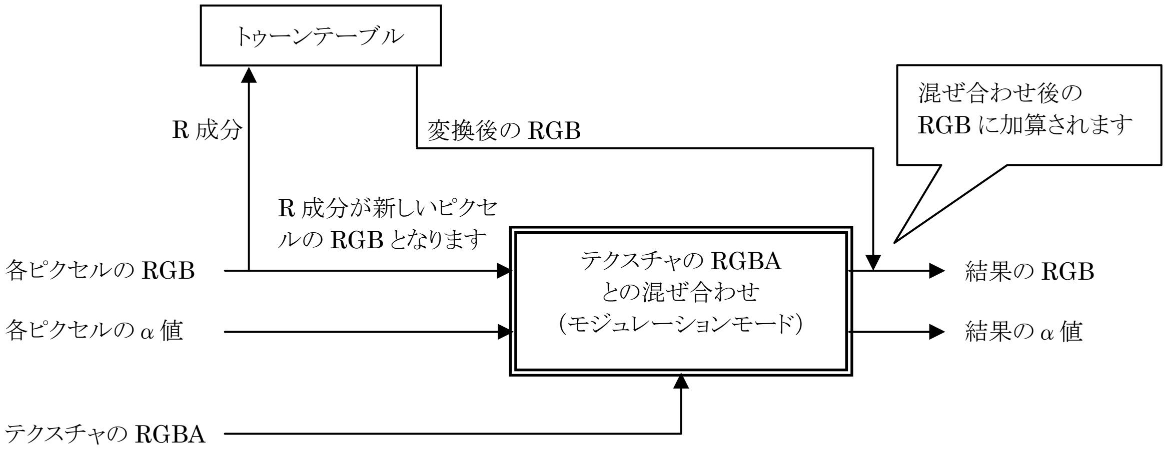


図 4-27　ハイライトシェーディング

【注意】

トゥーンシェーディングとハイライトシェーディングは、ハードウェアの仕様により、描画フレーム単位でどちらか一方しか選択することができません。つまり、同一描画フレーム中の、あるポリゴンにはトゥーンシェーディングを適用し、別のあるポリゴンにはハイライトシェーディングを適用するということはできません。（図 4-28）

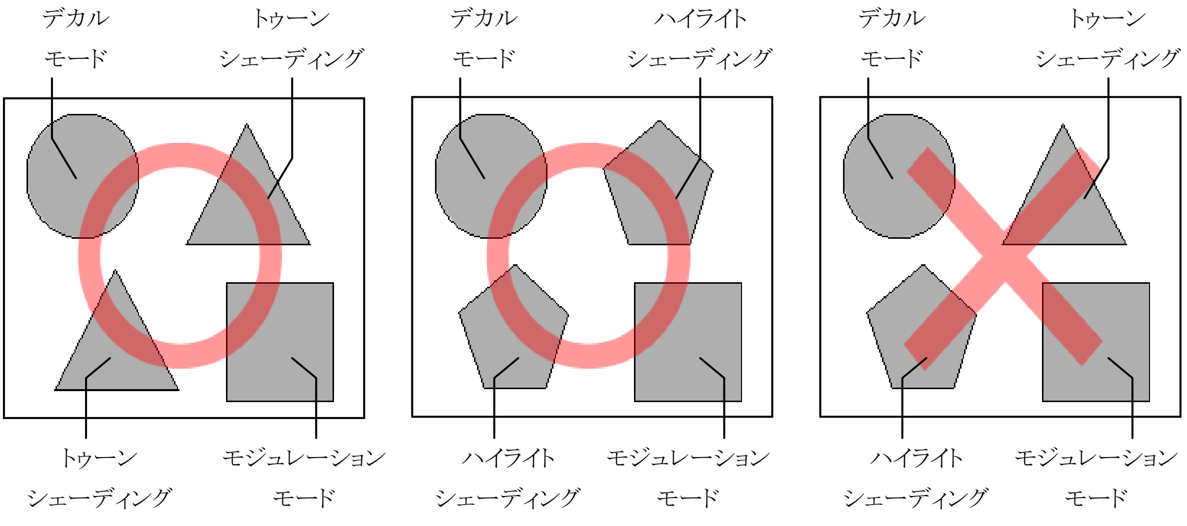


図 4-28 トゥーンシェーディングとハイライトシェーディングの注意事項

4.3.5 αテスト

αテストは、ピクセルのα値が指定した比較用α値以下の場合は、そのピクセルを描画しないようにする機能です。

αテストには、テクスチャとの混ぜ合わせ後のピクセルのα値が使用されます。（図 4-29）

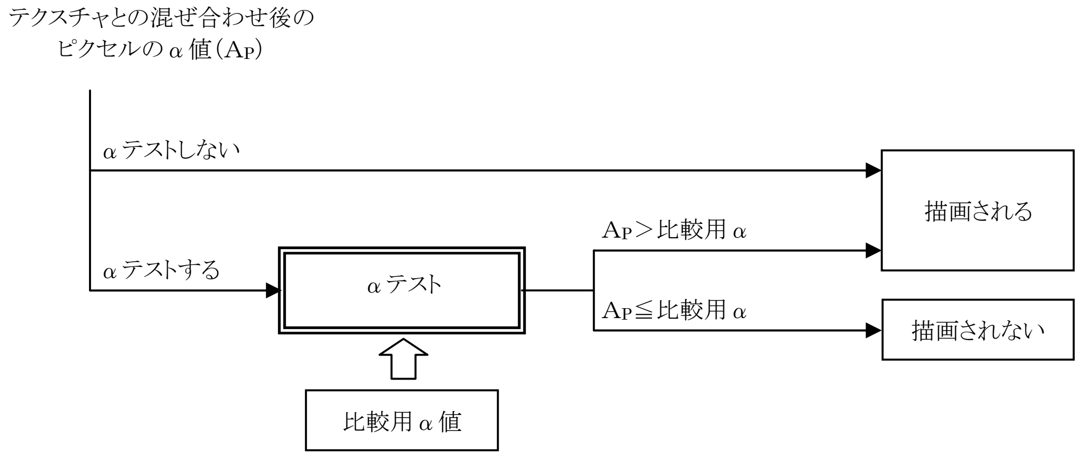


図 4-29 αテスト

4.3.6 αブレンディング

αブレンディングは、ピクセルのRGBとカラーバッファのRGBを、ピクセルのα値を使用して混ぜ合わせ、カラーバッファに描き込む機能です。（図 4-30）

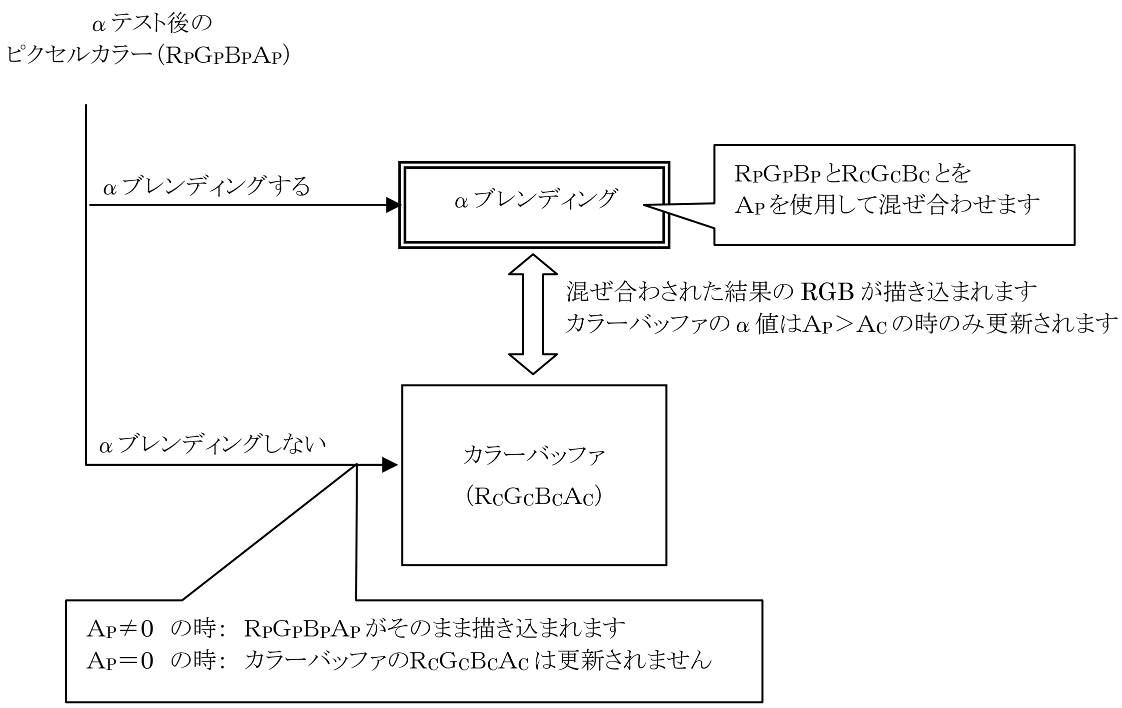


図 4-30 αブレンディング

4.3.7 エッジマーキング

エッジマーキングは、同じポリゴン ID を持つ不透明ポリゴン同士の輪郭線を描画する機能です。

輪郭線の色は任意に設定した 8 色から選択することができます。（図 4-31）

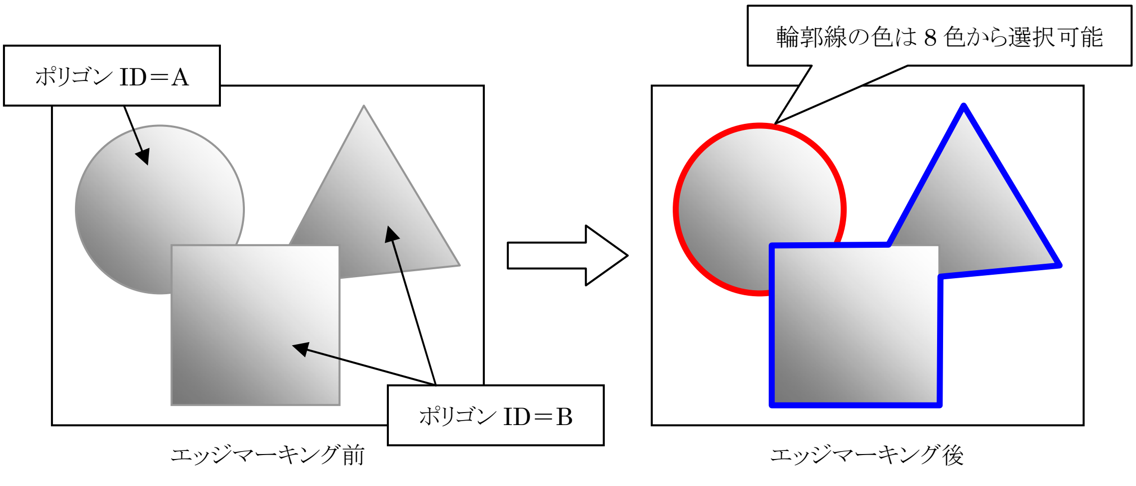


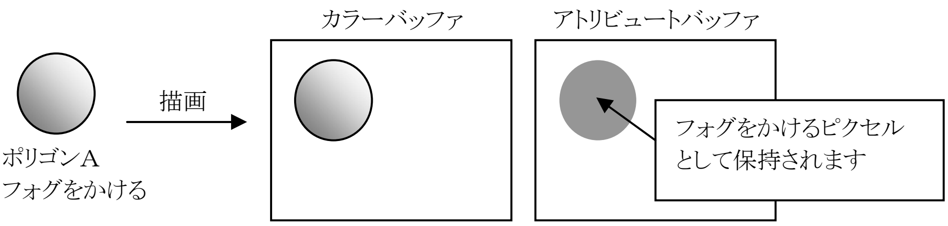
図 4-31 エッジマーキング

4.3.8 フォグブレンディング

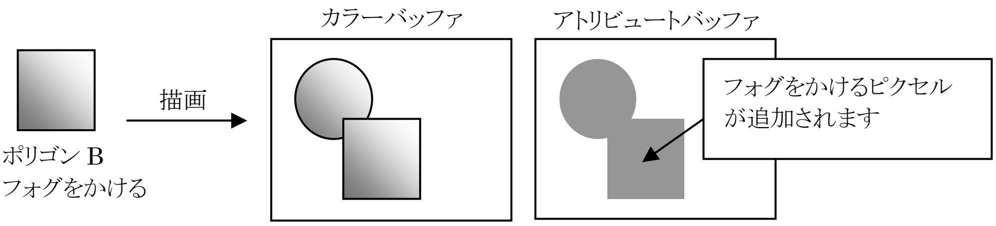
ポリゴンを描画していく際に、ポリゴン毎にフォグをかける／かけないを指定します。

「フォグをかける」と指定したポリゴンについては、その情報がアトリビュートバッファと呼ばれる専用のバッファに保持されます。そして、全てのポリゴンが描画された後で、カラーバッファの該当するピクセルに対して、フォグカラーの混ぜ合わせ処理が行われます。（図 4-32）

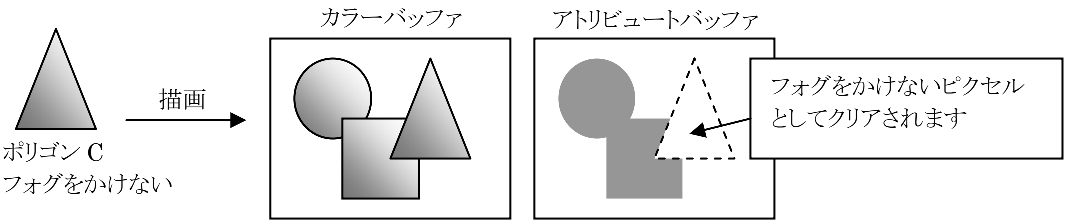
1) フォグをかけるポリゴン A を描画
アトリビュートバッファには、ポリゴン A の領域が保持されます。



2) フォグをかけるポリゴン B を描画
アトリビュートバッファには、ポリゴン B の領域が追加されます。



3) フォグをかけないポリゴン C を描画
アトリビュートバッファには、ポリゴン C の領域がフォグをかけないピクセルとしてクリアされます。



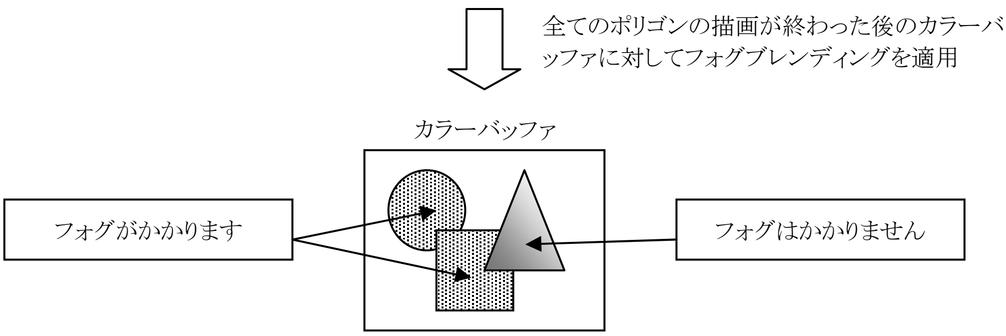


図 4-32 フォグブレンディングの流れ

■　フォグに設定できるパラメータ

フォグの計算には以下のパラメータを設定することができます。

- フォグカラー・アルファ（RGBA）
- フォグモード
- フォグ濃度テーブル

○　フォグカラー・アルファ（RGBA）

フォグのカラーとアルファを指定します。

○　フォグモード

フォグモードはフォグの混ぜ合わせ方を決めるモードで、次の 2 種類があります。

- カラーバッファの RGBA に対してフォグの RGBA を混ぜ合わせる
- カラーバッファの RGB はそのままで、カラーバッファの α 値に対してのみフォグの α 値を混ぜ合わせる

○　フォグ濃度テーブル

フォグ濃度テーブルは、ピクセルのデプス値からフォグの濃度を決めるためのテーブルです。

フォグ濃度テーブルにより、フォグのかかり方を制御することができます。

4.3.9　アンチエイリアス

アンチエイリアスは、描画されたポリゴンの境界部に対して、隣接するピクセル（上下左右の 1 ピクセル）との色の混ぜ合わせを行うことにより、ポリゴンの境界部のジャギー（ギザギザ感）を減らす機能です。（図 4-33）

アンチエイリアスをかける際の色の混ぜ合わせ方は、ハードウェアでその計算方法が決められているため、ソフト側での制御はできません。

アンチエイリアスの詳細については、TWL プログラミングマニュアルの該当章を参照してください。

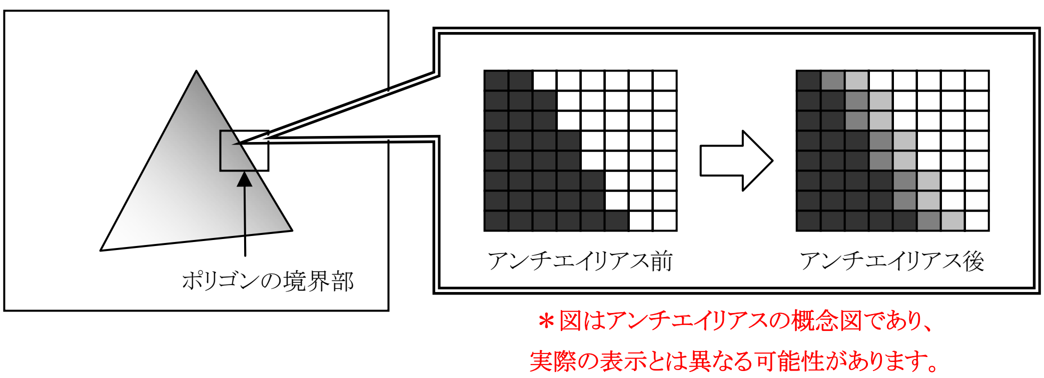


図 4-33　アンチエイリアス

4.3.10 補足 2：不透明ポリゴンと半透明ポリゴンの描画順序

RE で扱われるポリゴンは、不透明ポリゴンと半透明ポリゴンに分類されます。

RE で扱われるポリゴン
- 不透明ポリゴン
- 半透明ポリゴン

不透明ポリゴンと半透明ポリゴンの分類は以下のように行われます。（図 4-34）

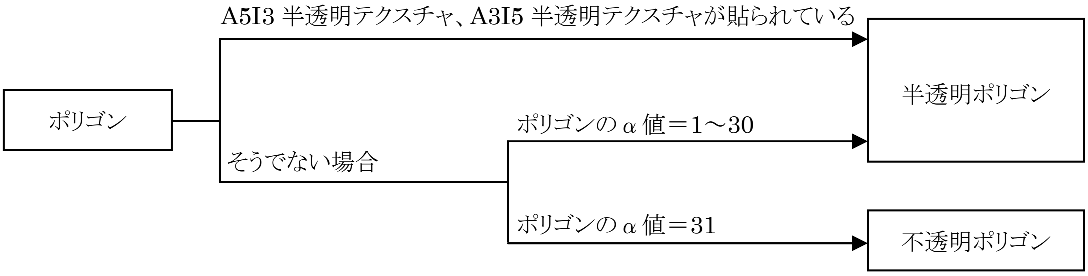


図 4-34 不透明ポリゴンと半透明ポリゴンの分類

分類された不透明ポリゴンと半透明ポリゴンは、必ず不透明ポリゴン→半透明ポリゴンの順に描画されます。（図 4-35）

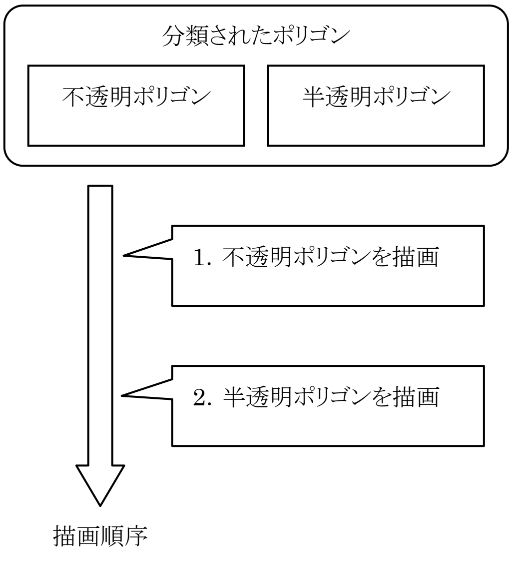


図 4-35 不透明ポリゴンと半透明ポリゴンの描画順序

4.3.11 補足 3：シャドウポリゴン

ポリゴンに影を落とすための手段として、シャドウボリューム法を用いたハードウェアによるシャドウポリゴン描画機能をサポートしています。シャドウボリューム法とは、ライトの光をポリゴンが遮る領域（影になる空間領域）をシャドウポリゴンとして用意し、シャドウポリゴンと接地面との交わりから影になる領域を決める手法です。（図 4-36）

RE は、シャドウポリゴンに指定されたポリゴンを描画する際に、内部に保持しているステンシルバッファを使用して、影の領域となる形を作成し、描画します。ステンシルバッファは、シャドウポリゴン機能のみに使用される専用のバッファです。

なお、シャドウポリゴンに指定するシャドウボリュームとなるポリゴンは、ハードウェアで自動的に生成してくれるわけではありませんので、ソフト側で用意する必要があります。

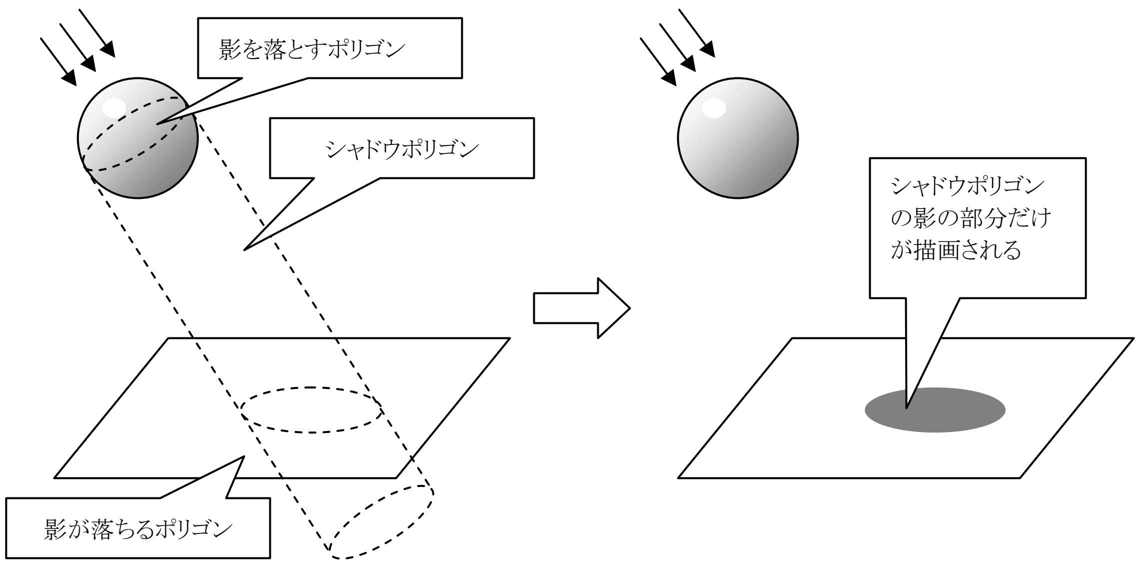


図 4-36 シャドウポリゴン

4.3.12 補足 4:カラーバッファとデプスバッファの初期化

カラーバッファとデプスバッファは描画フレーム毎に初期化されます。

カラーバッファとデプスバッファの初期化には、以下の 2 種類の初期化方法があります。

- 値で初期化する
- イメージデータで初期化する

○ 値で初期化する

カラーバッファとデプスバッファを任意のカラー値(RGBA)とデプス値で一様に初期化します。

○ イメージデータで初期化する

カラーバッファとデプスバッファを VRAM 上に用意したカラーイメージデータとデプスイメージデータで初期化します。

5 2D面と3D面の融合

5.1 2D面と3D面の合成

3Dグラフィックスでレンダリングされた画像は、ハードウェアの仕様により2DグラフィックスエンジンAのBG0 面に描画されます。2DグラフィックスエンジンAのBG0 面に描画された 3Dグラフィックス画像は、2DグラフィックスエンジンAの他のBG面やOBJと重ね合わせることで、2D面と3D面を合成して表示することができます。(図 5-1)

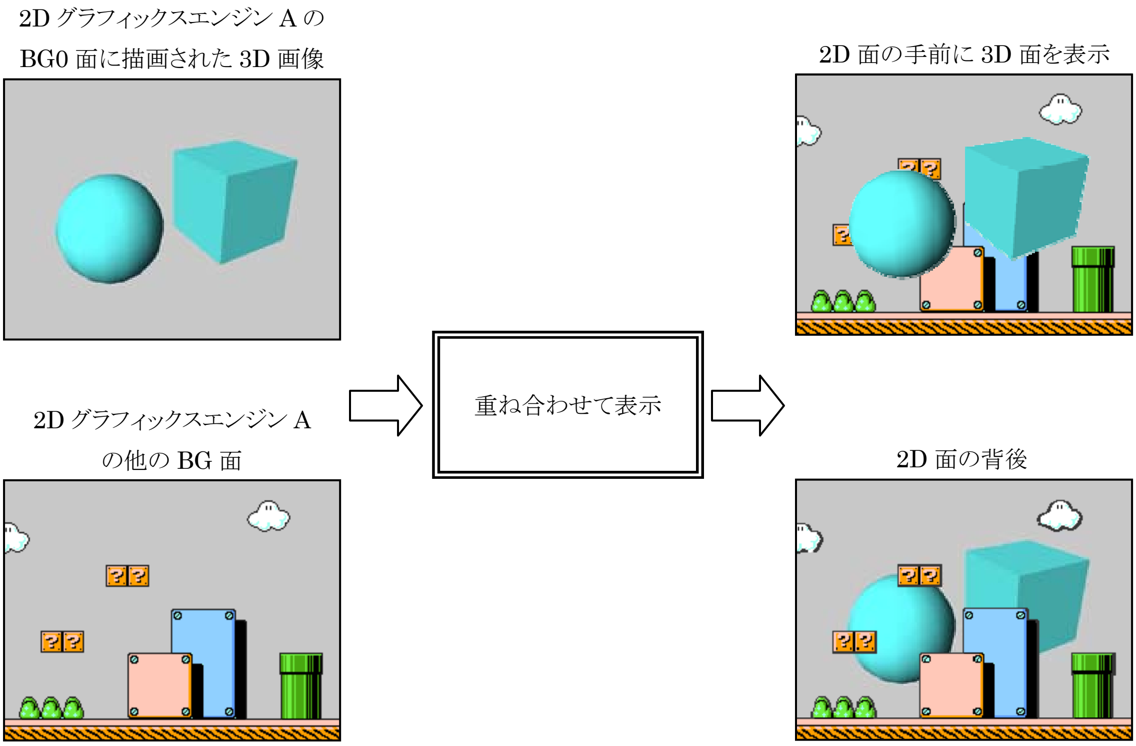


図 5-1 2D 面と 3D 面の合成

5.2 2Dグラフィックス機能を 3D面に適用する

2DグラフィックスエンジンAのBG0 面に描画された 3Dグラフィックス画像に対しては、以下の 2Dグラフィックス機能を適用することができます。（図 5-2）

- ラスタスクロール（ただし水平方向のみ）
- 2D グラフィックスエンジン A の他の BG 面との優先順位の指定
- ウィンドウの適用
- カラー特殊効果



図 5-2 2D グラフィックス機能を 3D 面に適用する

5.3 3Dグラフィックス機能を 2D面に適用する

3Dグラフィックスのフォグ機能を、3D面と合成された 2DグラフィックスエンジンAの 2D面にもかけることができます。（図 5-3）

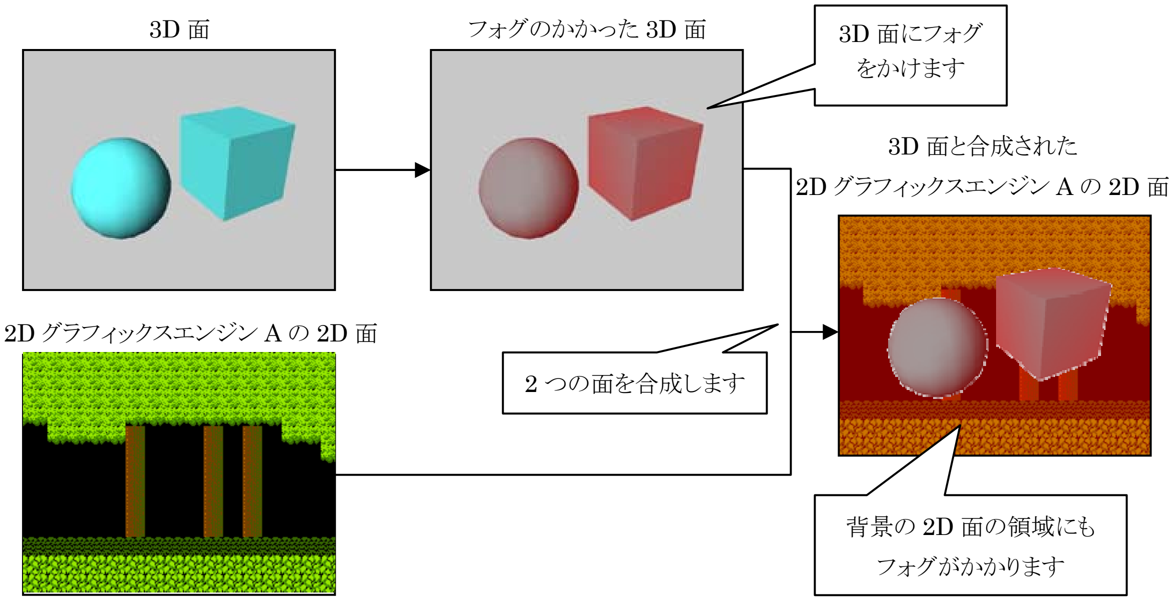


図 5-3 3D グラフィックス機能を 2D 面に適用する

【注意】

本章で解説している 2D 面と 3D 面の融合機能は、2D グラフィックスエンジン B には適用することができません。

6 表示キャプチャ

表示キャプチャ機能とは、2D グラフィックスエンジン A の出力画像、3D グラフィックスの出力画像、VRAM 上の画像データ、メインメモリ上の画像データを、LCD 画面に表示せずに、VRAM 上に画像データとして取り込む機能です。

この機能の流れは、次のようになります。（図 6-1）

① キャプチャ元画像 A を以下の 2 種類から選択します。

- 2D グラフィックスエンジン A による 2D／3D グラフィックス合成後の結果の画像
- 3D グラフィックスの出力画像

② キャプチャ元画像 B を以下の 2 種類から選択します。

- VRAM 上の画像データ
- メインメモリ上の画像データ

③ ①, ②で選択したキャプチャ元画像に対するキャプチャ方法を、以下の 3 種類から選択します。

- キャプチャ元画像 A をキャプチャする
- キャプチャ元画像 B をキャプチャする
- キャプチャ元画像 A, B を混ぜ合わせてキャプチャする（混ぜ合わせ比率を指定することができます）

④ ③で選択したキャプチャ方法でキャプチャされた画像データが VRAM 上に格納されます。

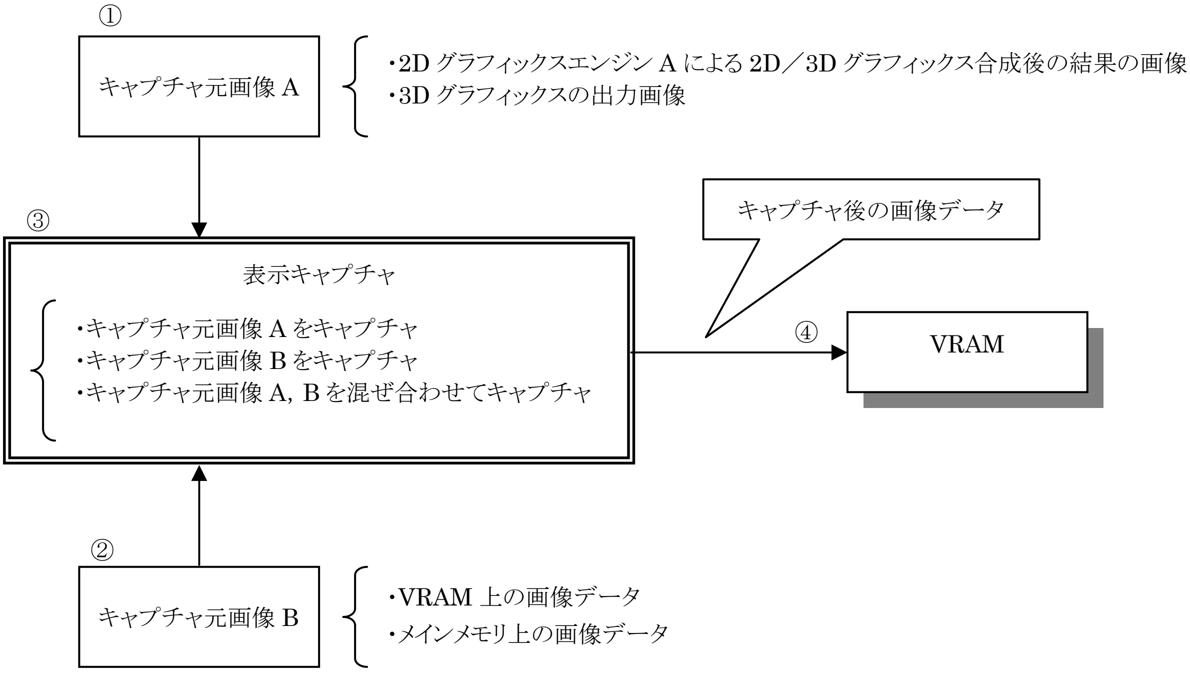


図 6-1 表示キャプチャ機能

■ 表示キャプチャ機能の使用例

表示キャプチャ機能の使用例として、3Dグラフィックスの出力画像をキャプチャし、2DグラフィックスエンジンAのOBJとして表示する例を以下に示します。(図 6-2)

① 3D 画像をレンダリングします。

② 表示キャプチャ機能で 3D 画像を VRAM に取り込みます。

③ 取り込んだ 3D 画像を、次の描画フレーム以降で、OBJ として 2D グラフィックスエンジン A で使用します。

④ 2D グラフィックスエンジン A で BG と OBJ を重ねて表示します。

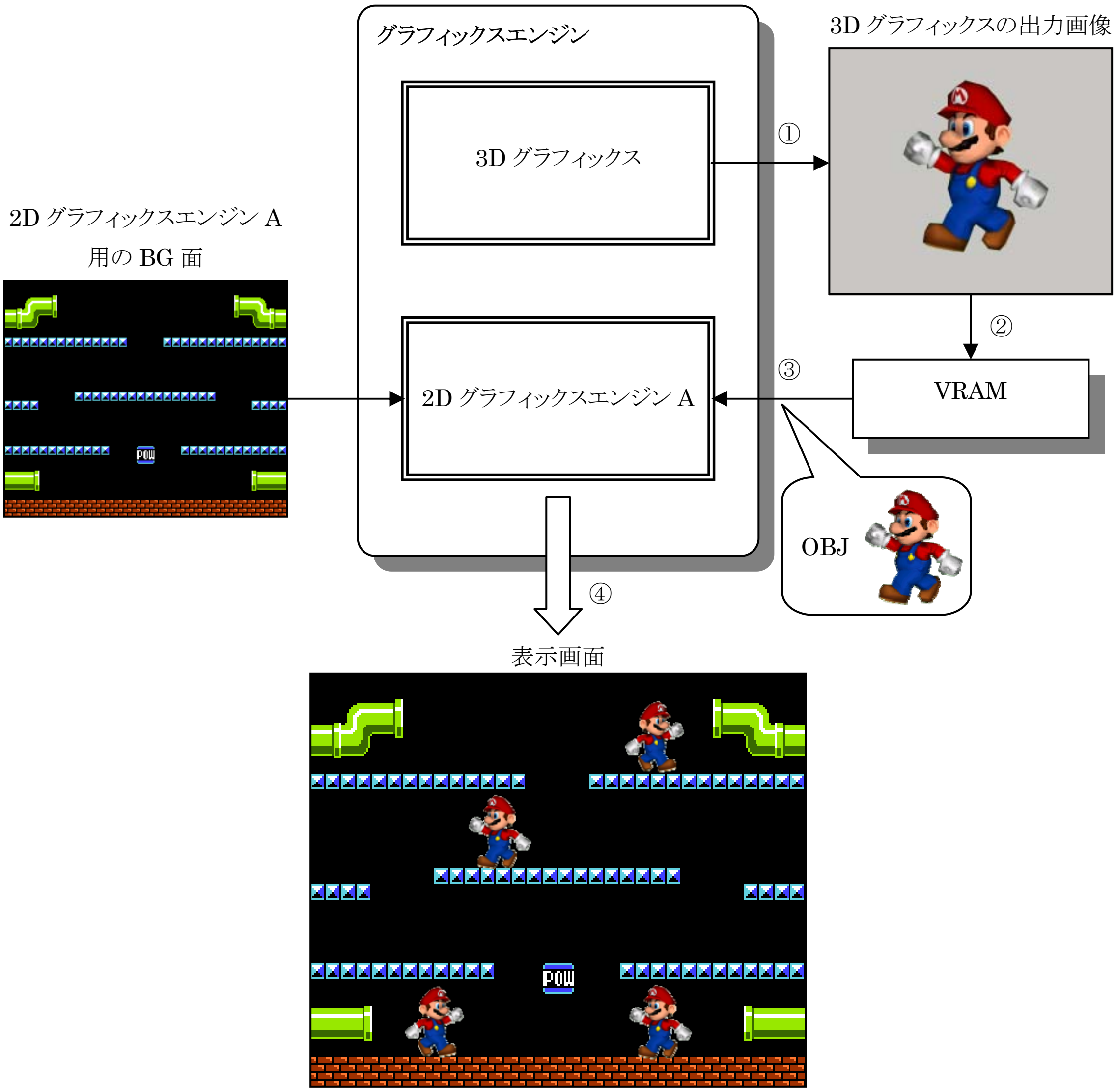


図 6-2 表示キャプチャ機能の使用例

【注意】

本章で解説している表示キャプチャ機能は、2D グラフィックスエンジン B には適用することができません。

7 付録

付録では、前章までに説明してきた各グラフィックス機能を使用して、ハードウェア全体としてどのようなことができるかについて例を挙げて説明します。

7.1 グラフィックス表示モード使用例 1

図 7-1は、グラフィックス表示モードを使用して 2D画像と 3D画像を重ね合わせて表示する例です。3Dグラフィックスエンジンによって生成された 3D画像は、2DグラフィックスエンジンAのBG0 面に描画され、他のBG面やOBJ面と重ね合わせて表示することができます。

■ 図の説明

① 3D グラフィックスエンジンによって生成された 3D 画像は、2D グラフィックスエンジン A の BG0 面に描画されます。

② 3D 画像が描画された BG0 面と、他の BG 面および OBJ 面との重ね合わせが実行され、LCD に表示されます。

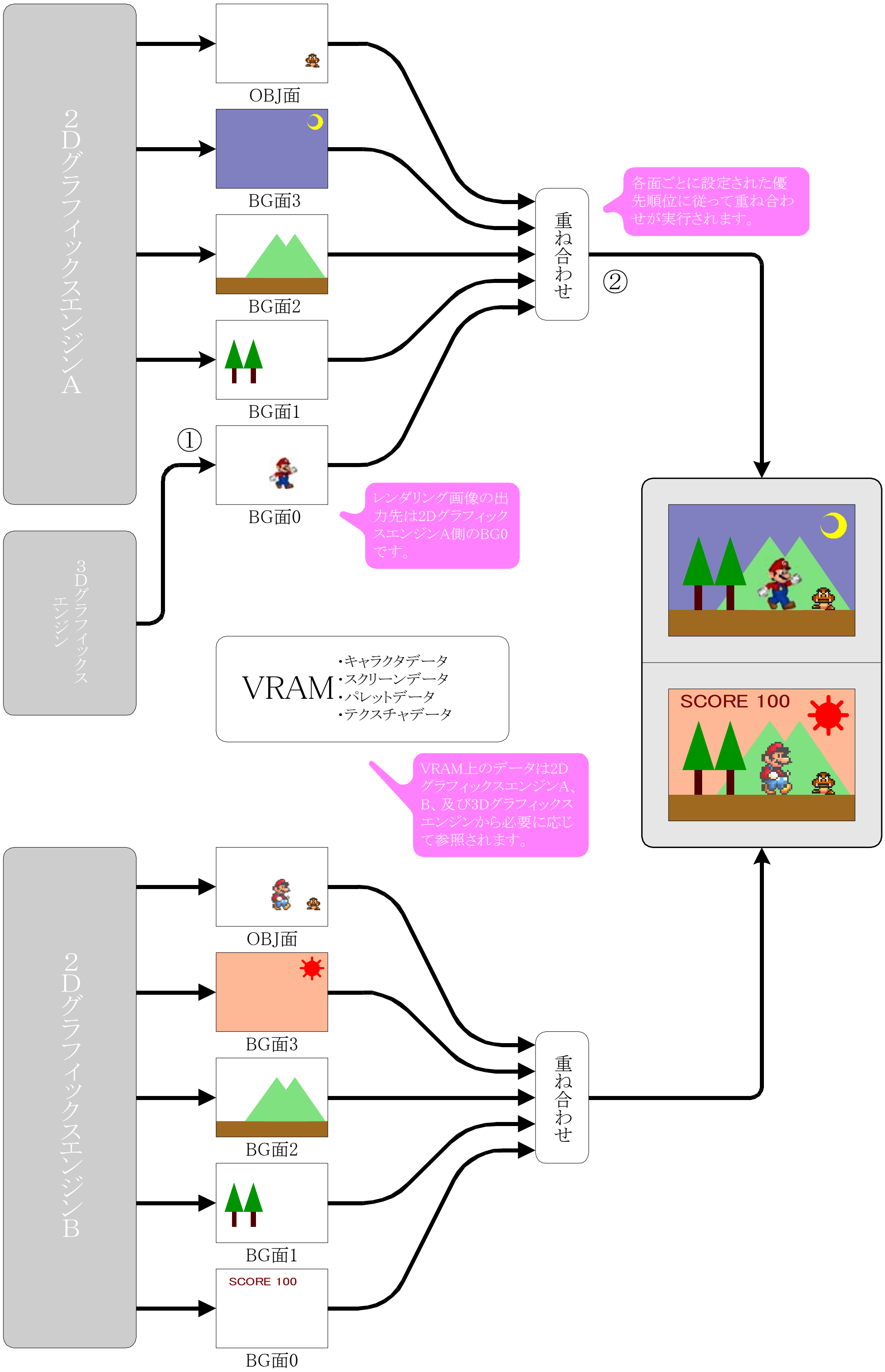


図 7-1 グラフィックス表示モード使用例 1

7.2 グラフィックス表示モード使用例 2 ＜キャプチャ機能＞

図 7-2は、キャプチャ機能の使用例です。3Dグラフィックスエンジンによって生成された 3D画像をキャプチャ機能によってVRAMに取り込み、ビットマップOBJとして使用しています。

■ 図の説明

① 3D グラフィックスエンジンによって 3D 画像を生成します。

② 生成された 3D 画像をキャプチャ機能で VRAM に取り込みます。

③ VRAM に取り込まれた画像データは、次の表示フレーム以降にビットマップ OBJ として使用できます。

■ プログラマー向けの話

グラフィックス表示モードでは、キャプチャしたデータを書き込む VRAM と OBJ の表示に用いる VRAM を別に用意しなければなりません。キャプチャ用と OBJ 表示用の 2 つの VRAM を使用して、フレーム毎に役割を入れ替えるといった工夫が必要になります。

また、2D グラフィックエンジン A と B は、OBJ 表示用の VRAM をそれぞれ別に持つ必要があります。従って、2Dグラフィックスエンジン A に割り当てられた VRAM にキャプチャした画像を、グラフィックスエンジン B でも同時に使用したい場合は、グラフィックスエンジン B に割り当てられた VRAM へ画像をコピーしておく必要があります。

例ではキャプチャした画像をビットマップ OBJ として使用しましたが、ビットマップ BG として使用することもできます。

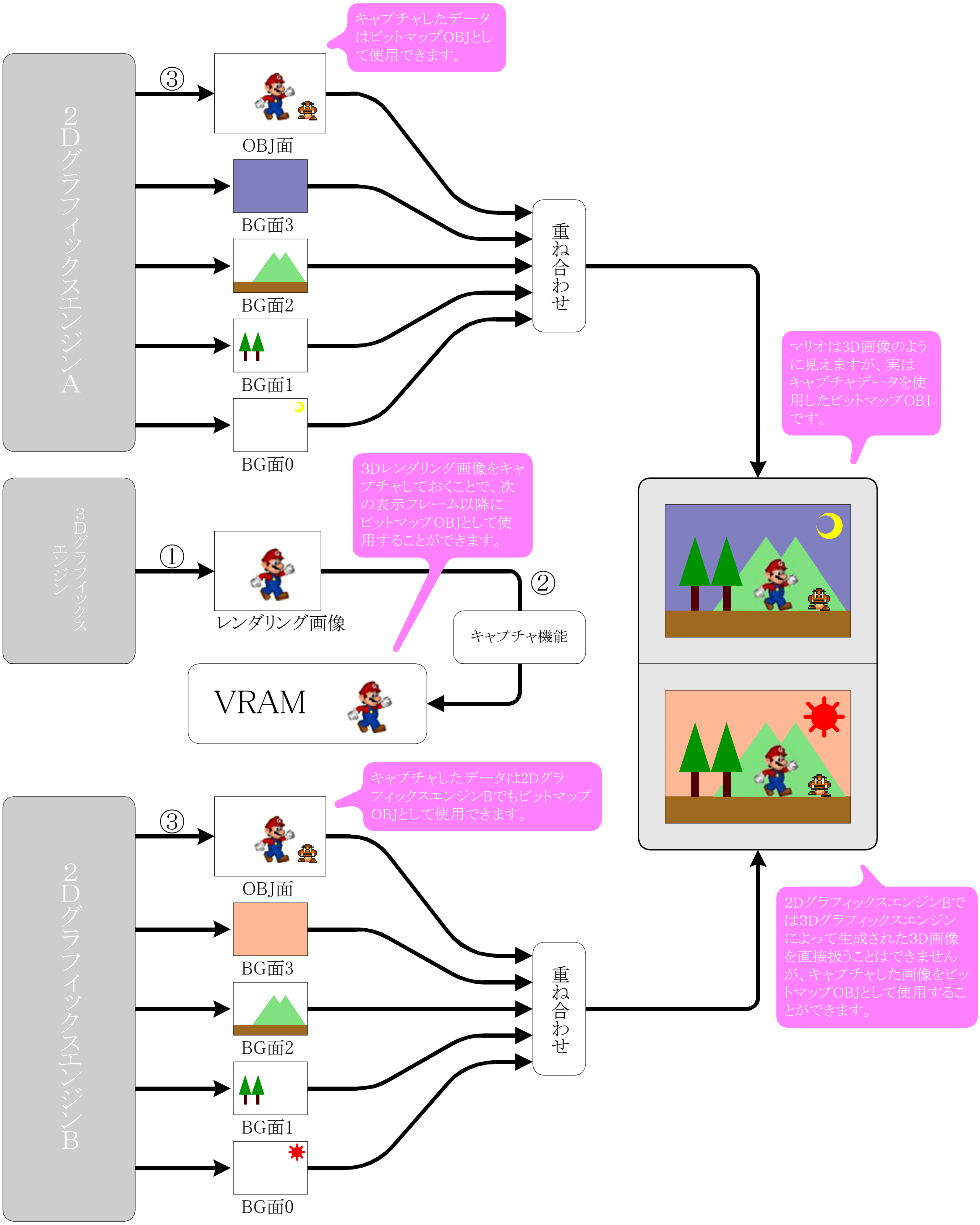


図 7-2 グラフィックス表示モード使用例 2<キャプチャ機能>

7.3 グラフィックス表示モード使用例 3 ＜2 画面での 3D表示＞

DS には 3D グラフィックスエンジンが 1 つしか実装されていないため、メイン LCD とサブ LCD の両方に同時に 3D 画像を表示することはできません。しかし、3D グラフィックスエンジンの出力先を 1 フレーム毎に切り替えることにより、(1 フレーム置きの更新となりますが) 両方の LCD に 3D 画像を表示することができます。

例では、メインLCDに通常のマリオを、サブLCDにルイージを登場させています。図 7-3は奇数フレームの処理を、図 7-4は偶数フレームの処理を表しています。

■ 図の説明 (奇数フレーム)

① 3D グラフィックスエンジンでマリオをレンダリングします。

② 次の表示フレームに備えて、2D 面と 3D 面を重ね合わせた画像を VRAM-D にキャプチャします。

③ メイン LCD 側は、出力を 2D グラフィックスエンジン A の VRAM 表示モード (VRAM-D) にすることで、キャプチャと同時に前前回の表示フレームで VRAM-D にキャプチャしておいた画像を表示します。

④ サブ LCD 側は、前回の表示フレームにキャプチャしておいた VRAM-C 上の画像をビットマップ BG として表示します。

■ 図の説明 (偶数フレーム)

① 3D グラフィックスエンジンでルイージをレンダリングします。

② 次の表示フレームに備えて、2D 面と 3D 面を重ね合わせた画像を VRAM-C にキャプチャします。

③ サブ LCD 側は、出力を 2D グラフィックスエンジン A の VRAM 表示モード (VRAM-C) にすることで、キャプチャと同時に前前回の表示フレームで VRAM-C にキャプチャしておいた画像を表示します。

④ メイン LCD 側は、前回の表示フレームにキャプチャしておいた VRAM-D 上の画像をビットマップ OBJ として表示します。

■ プログラマー向けの話

奇数フレームでは VRAM-C 上の画像を BG として表示しているのに対し、偶数フレームでは VRAM-D 上の画像を OBJ として表示していますが、これには理由があります。
DS の仕様として、2D グラフィックスエンジン B に VRAM-C を割り当てた場合、用途は BG データ参照用に限られます。同様に、2D グラフィックスエンジン B に VRAM-D を割り当てた場合、用途は OBJ データ参照用に限られます。このような仕様のために BG と OBJ を使い分けています。

このアルゴリズムでは、VRAM 表示モードの VRAM とキャプチャ先の VRAM が同じ場合にはキャプチャ前の VRAM の内容で描画が行われることを利用しています。

奇数フレーム

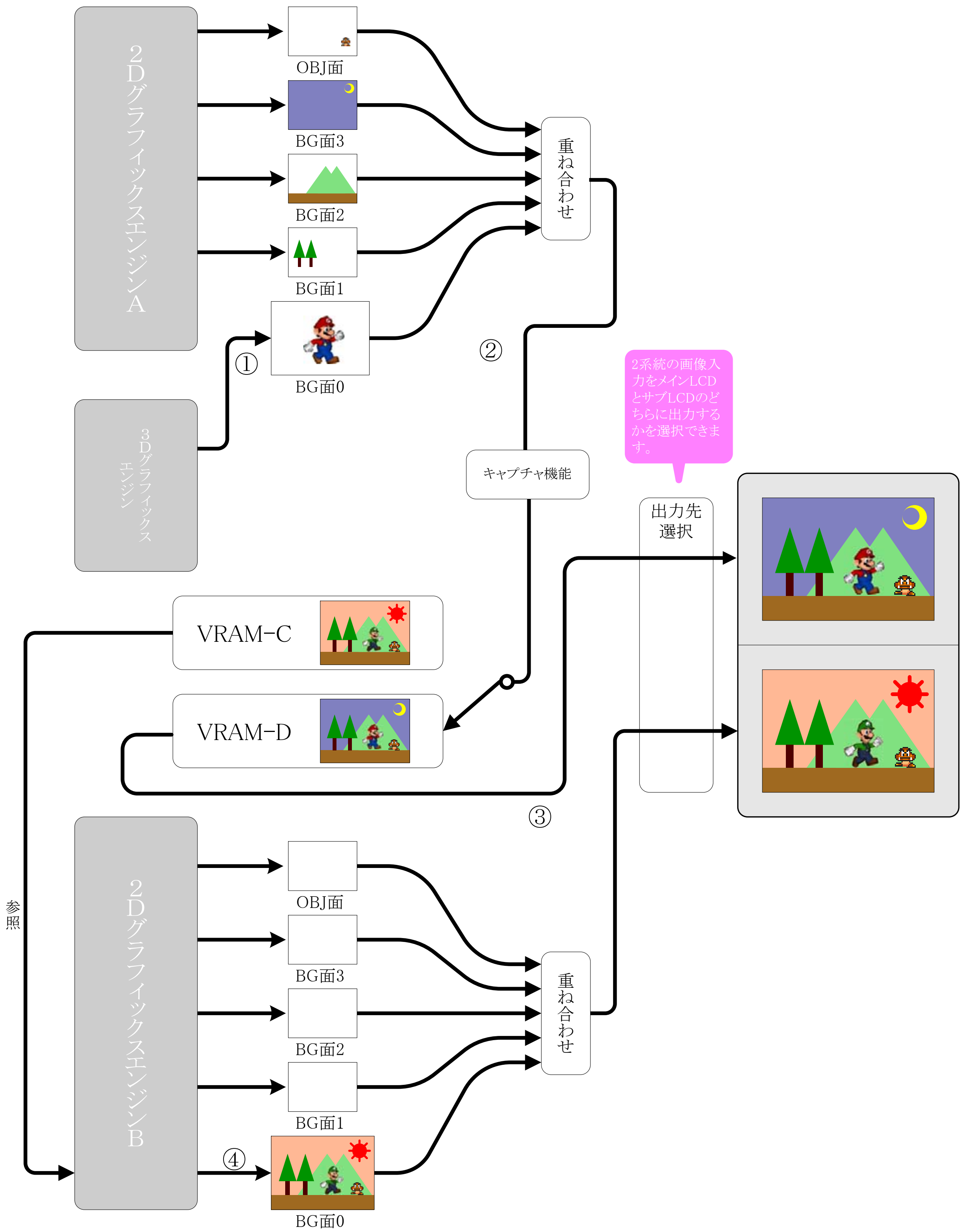


図 7-3 メイン LCD とサブ LCD の両方で 3D 表示を行う例（奇数フレーム）

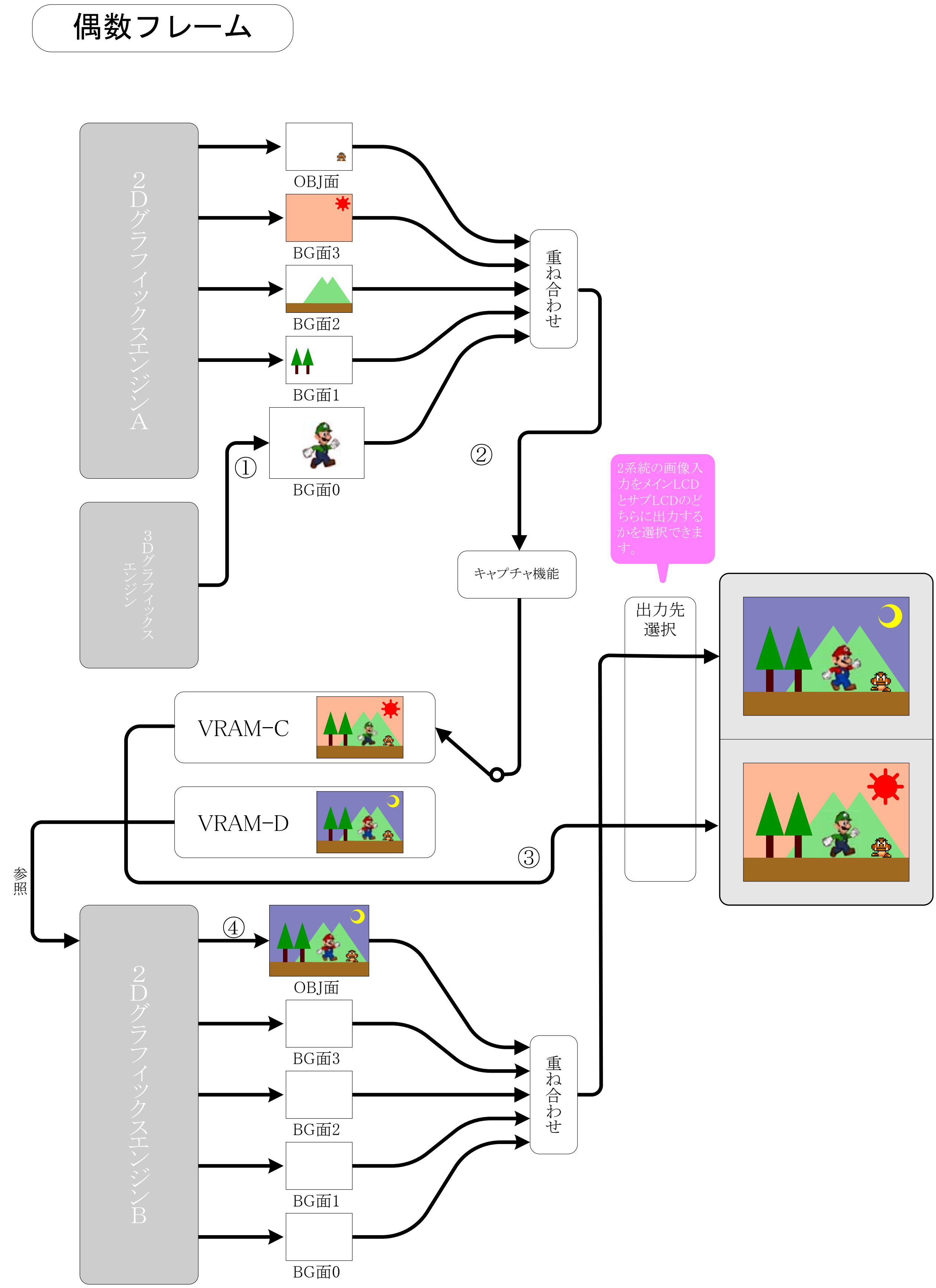


図 7-4 メイン LCD とサブ LCD の両方で 3D 表示を行う例（偶数フレーム）

7.4 VRAM表示モード使用例 ＜モーションブラー効果＞

VRAM表示モードでは、VRAM上にある 1 フレーム分のビットマップデータを表示することができます。図 7-5の例では、グラフィックス回路により生成した画像と現在表示している画像をブレンドし、その結果をキャプチャ機能でVRAMに取り込むことによってモーションブラー効果を実現しています。

■ 図の説明

① VRAM 上のビットマップデータは LCD に表示されます。

② LCD に表示する画像をキャプチャ元 B に設定します。

③ 新たにグラフィックス回路で生成した画像をキャプチャ元 A に設定します。

④ キャプチャ元 A とキャプチャ元 B の画像をブレンドします。

⑤ ブレンド後の画像をキャプチャ機能で VRAM に取り込みます。取り込んだ画像は次の表示フレームに表示します。

■ プログラマー向けの話

VRAM 表示モードでは、表示に使用している VRAM に対してキャプチャした画像を書き込むことができます。

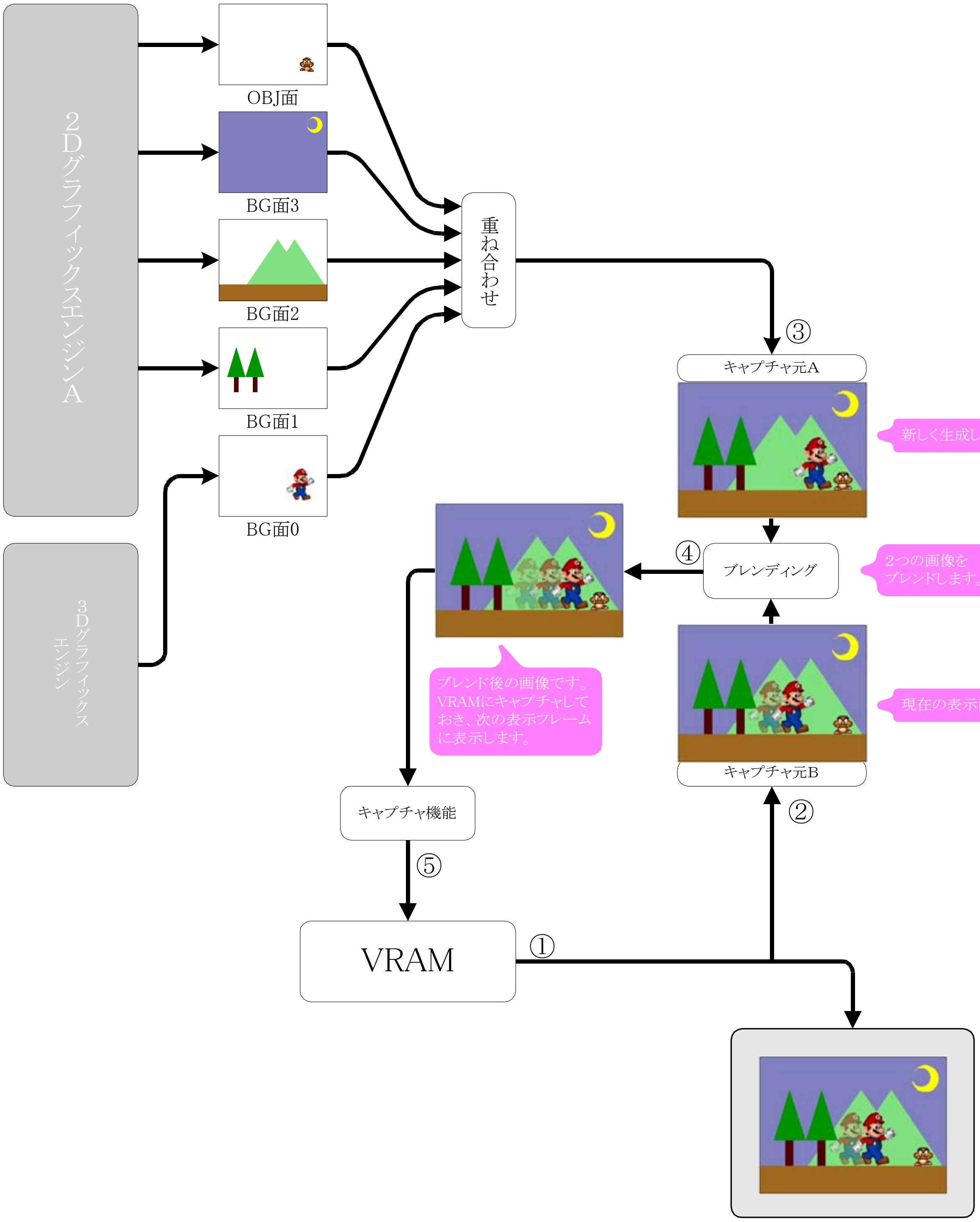


図 7-5 VRAM 表示モード使用例＜モーションブラー効果＞